

3-2011


Case Study Evaluation of Dynamic Traffic Assignment Tools

John Gliebe
Portland State University

Åsa Bergman
Portland State University

Let us know how access to this document benefits you.

Follow this and additional works at: http://pdxscholar.library.pdx.edu/usp_fac

 Part of the [Transportation Commons](#), [Transportation Engineering Commons](#), and the [Urban Studies Commons](#)

Citation Details

Interagency Agreement 24620. Work Order Contract #5. Produced by the Center for Urban Studies, Portland State University, for the Oregon Department of Transportation, Planning Analysis Unit.

This Report is brought to you for free and open access. It has been accepted for inclusion in Urban Studies and Planning Faculty Publications and Presentations by an authorized administrator of PDXScholar. For more information, please contact pdxscholar@pdx.edu.

CASE STUDY EVALUATION OF DYNAMIC TRAFFIC ASSIGNMENT TOOLS

Interagency Agreement 24620
Work Order Contract #5

John P. Gliebe
Åsa E. Bergman

Center for Urban Studies
Portland State University
P.O. Box 751 USP
Portland, OR 97207-0751

for

Oregon Department of Transportation
Transportation Planning Analysis Unit
555 13th St SE, Suite 2
Salem, Oregon 97310

March 2011

Executive Summary

A case study was undertaken in order to evaluate the potential use of dynamic traffic assignment (DTA) tools by Oregon Department of Transportation (ODOT) and its partner agencies. The objectives of this study were to provide insight into the nature of DTA models, to inform the program selection process, and to develop realistic expectations for potential DTA work plans. The overarching goal of this report is to describe the process followed and experiences of the study team in developing and testing DTA network models.

Two available DTA programs were selected for in-depth analysis from a preliminary screening of available programs: DynusT, an open-source program, and Dynameq, a commercially licensed product of INRO Inc. DynusT is a mesoscopic simulation DTA, with link-level resolution. In contrast, Dynameq is a mesoscopic simulation DTA with microscopic elements, such as lane-level resolution and lane-changing behavior.

A case study area was created from the Portland regional model maintained by Metro. The study area was centered on the City of Beaverton and included portions of US 26 and OR 217. The network model consisted of 106 zones, 630 nodes, 1560 links, and 136 signalized intersections.

A four-stage development process was undertaken to gradually develop DTA models in both DynusT and Dynameq. The first step was to import a sub-area network from an existing Metro planning model program. A set of trip tables representing various periods of the day was allocated to 15-minute time intervals using the temporal distribution of person travel from the 1994 Metro household survey. Available demand included single- and multiple-occupancy auto trips and both medium- and heavy-truck trips. In the initial stage, default fixed time signal timing plans were applied globally and only single-occupancy vehicle (SOV) trips were used as demand. A PM 3-hour peak period was tested and used to diagnose and correct problems related to network coding, unrealistic signal timing, and inaccurate loading of demand. Problems manifested themselves in the form of blocked intersections that would spread throughout the network to produce gridlock. When this happens, the network fails to “clear” and vehicles remain stuck in the network.

In the second stage, actual signal timing plans were acquired for the study area from multiple jurisdictions. The complexity of many of the area’s timing plans, which include semi-actuated dual-ring controls with pedestrian actuation and, in some corridors, synchronization, could not be represented directly in either DTA program. Simplifications were required, resulting in fixed-time plans being coded, with synchronization provided where appropriate. These timing plans were implemented, enabling networks with SOV demand to clear. In the third stage of development, the remaining autos and trucks were added to account for 100 percent of demand. Additional problems surfaced related to intersection capacities and centroid connector demand and placement. While many problems were solved, the study team was unable to get a 100 percent PM peak scenario to clear and settled for a 75 percent scenario.

In a fourth stage, models were run for AM Peak, Midday and 24-hour scenarios. The AM Peak and Midday scenarios cleared in both programs, using full demand and signal timing plans.

Despite several attempts, the 24-hour scenario could not be run successfully with signalization and full demand. Only a 75 percent-demand 24-hour scenario without signalization would clear.

The Midday scenario was chosen for subsequent analysis and testing due to its ability to clear with full demand and signalization. Tests to determine how much more demand could be accommodated in the Midday scenario revealed that the network could handle only about 10 percent more demand. Other sensitivity tests compared the performance of each program under realistic, default and fully actuated signal timing settings. Additional sensitivity tests considered the impacts of freeway lane closures and realistic versus arbitrary turning bay lengths.

For each of these scenarios, the two programs generated aggregate network statistics related to vehicles clearing the network, average travel and stop times and trip distance, closure criteria, and processing time. As a general trend, Dynameq produced longer trip distances but shorter travel times. Dynameq also required significantly less CPU time (36-70%) for the same number of iterations and, in most cases, achieved a better relative gap (measure of convergence). One obvious difference between the two programs is that DynusT assigns intra-zonal trips to the network, and Dynameq does not. Intra-zonal trips added 2 percent more vehicles to a DTA in this study, and those trips tended to be shorter and use slower arterials. An additional factor is that Dynameq seemed to assign more vehicles to freeways compared with DynusT, and those trips tended to be longer in distance but shorter in travel time.

In addition, both programs produce a wealth of output related to the performance of individual network links, such as estimated volumes, densities, speeds, delay, and travel times. Both programs also save the paths of all vehicles. Scenario volumes and speeds were compared with count data for 46 arterial sites and count and speed data for 18 freeway detector locations. Comparing the ratios of estimated-to-observed volumes by facility type and hourly time period, it was found that DynusT would come closer to the counts overall, although Dynameq would occasionally perform better on freeways.

Where the two programs seemed to differ most was on speeds. Comparisons with 18 freeway detector locations demonstrated that Dynameq would produce relatively stable speed estimates, more so than the observed speeds, whereas DynusT would consistently produce more dramatic fluctuations than the observed speeds. This last observation probably explains most of the differences in network-wide average travel times between the two programs. Further, the sensitivity test on a freeway lane closure, revealed very different response patterns between the two programs, with Dynameq responding in what appeared to be a more realistic way and DynusT slowing traffic to a crawl. While it seems that in these limited tests Dynameq was finding better (shorter) paths for vehicles and converging faster, the fact that DynusT tended to better match traffic counts suggests that real drivers might be less aware of shortest travel time paths than the models portray.

It is important to remember that the neither program was calibrated in this study, and that parameter tuning would be expected to improve the implied behaviors and performance of both models. Calibration involves a much more detailed investigation of network choke points using turning movement counts and select link analysis, followed by adjustments to individual link speed-flow-density parameters. Both program developers also recommended modifications to the

network zone system and centroid connectors as well as calibration of the time-dependent demand tables in order to better match observed demand patterns. Both programs also enable the user to modify driver reaction parameters.

In terms of user experience, both program vendors provide first-rate user support. The technical manuals available to the study from both vendors were comparable, although the Dynameq documentation seemed to be more complete. The study team found many of the error messages produced by the DynusT program to be uninformative, compared with those produced by Dynameq. In addition, the study team found the graphic user interface of Dynameq easier to use.

Table of Contents

Executive Summary	1
Table of Contents	4
Background	6
Dynamic traffic assignment (DTA)	6
Study objectives	7
Remainder of this report	7
Selection of Programs	9
DynusT	10
Dynameq	11
MATSIM	12
Case Study Setup	14
Study area	14
Network files	15
Demand data	16
Common database	17
DTA Network Model Development	18
Incremental approach	18
Stage 1. Basic network, default signal timing, partial PM Peak demand	19
Stage 2. Add realistic signal timing plans	24
Stage 3. Run with full demand, apply peaking factors	28
Stage 4. Run AM Peak, Midday, and 24-hour scenarios	31
Evaluating Model Results	32
Checking for network clearance	32
Relative gap	33
More DTA output	34
Validation data	34
Scenario Analysis	37
Baseline PM Peak -- 3 to 6 p.m.	38
Baseline Midday -- 10 a.m. to 1 p.m.	44
24-Hour -- 1 a.m. to 11 p.m.	49
Saturated Midday (110%)	52
Sensitivity tests using Baseline Midday -- 10 a.m. to 1 p.m.	55
Next Steps and Calibration	62

Evaluation of User Experience	63
Data file handling (input and output).....	63
User interface and error reporting.....	63
Documentation.....	64
Technical support.....	64
Considerations for Choosing a DTA Program.....	66
Appendix A: Signal Timing Conversion	68
References.....	75

Background

The Oregon Department of Transportation's (ODOT) Transportation Planning and Analysis Unit (TPAU)) is in need of a network modeling tool that can better represent within-day dynamics of traffic flow for urban and regional planning applications. This need is the result of increasing interest on the part of policy makers for scenario analyses of time/congestion-varying road pricing, emissions analysis, reliability analysis, work-zone management and phasing, ramp metering, corridor studies, and other applications in which time-sensitive traffic flows are important. Potential future analyses related to the need to model greenhouse gas reduction scenarios might include projecting time-of-day demand at electric vehicle charging stations, time-dependent bus operations, and time-of-day demand for parking management and pricing, all of which are likely to require time-sensitive network costs as inputs. A time-dependent network modeling tool would not only better represent congestion effects by time of day, but also enable analysts to more accurately obtain vehicle speed estimates and hot and cold vehicle running times for emissions analysis.

The current set of network assignment modeling tools are static in nature, essentially aggregating traffic flow into multiple-hour peak and off-peak periods. Static traffic assignment (STA) models do not reflect shifts in demand between peak and off-peak periods or within each of these periods, are not very accurate in terms of vehicle speed estimation, and cannot portray queuing or variation in travel times and volumes for reliability analysis.

Dynamic traffic assignment (DTA)

In recent years, a general class of network modeling tools, known as dynamic traffic assignment (DTA), have been developed by university researchers, federal government research labs, and some commercial software makers. DTA programs differ from static models by adding the dimension of time-of-day specificity to the network assignment process. This is made operational by slicing the assignment of demand into time intervals (e.g., 5 to 15 minutes) and tracking vehicle movements at even finer levels of resolution (e.g., 6 seconds). Whereas STA involves finding a least-cost (usually travel time) path between each origin-destination (OD) pair in a network, DTA finds the least cost path corresponding to a specific departure time. Whereas STA algorithms are designed to find a user-equilibrium (UE) solution in which the used paths for a given OD pair all have the same travel cost, DTA algorithms are designed to find a dynamic user equilibrium (DUE) in which the used paths connecting a given OD pair all have the same travel cost for the same departure time interval. In terms of outputs, DTA will produce different sets of vehicle paths for the same OD pair for different departure time intervals, compared with a single set of paths for STA. Moreover, DTA will produce traffic volume and speed time profiles by user-specified time intervals for individual facilities, whereas STA can only produce volume and speed estimates based on an hourly average for the assignment period.

The motivation for this study was that the existing set of known DTA programs does not offer a well-accepted approach for practical application as an urban and regional planning tool for large study areas. To date, DTA models have been used primarily for research studies, the main limitations to everyday use being unwieldy computer processing requirements and lengthy run times for large problem sizes. In addition, there exist several different methods for modeling traffic networks dynamically, and it is unclear which method provides a sufficient level of

behavioral realism at a reasonable cost. Finally, DTA represents a fundamental leap in analytical modeling capabilities, which means that technical staff will need to undergo training in its capabilities and implementation. With these considerations in mind, it is imperative that DTA programs selected for further study and eventual adoption undergo a rigorous testing and evaluation process to meet TPAU objectives for performance, ease of use and cost.

Currently available approaches to DTA may be grouped into three broad categories: analytical solutions, heuristic approximations to unique solutions, and large-scale disequilibrium microsimulations. All three approaches start with a set of demand trip tables, which are “discretized” by time slices (e.g., 5 to 15 minutes). Analytical DTA programs are considered to be the purest, based on highly-sophisticated mathematical programs which attempt to find a unique, dynamic user equilibrium solution to the problem of loading trips on the network with time-dependent travel costs. In contrast, there are several heuristic approaches which use iterative simulation methods in order to generate a solution to the dynamic network assignment problem, but do not guarantee a globally optimal solution. A third approach, regional microsimulation, is similar to familiar traffic engineering simulations, but on a regional scale, using car-following rules and usually a highway network “discretized” by space-time cells (e.g., the distance that a vehicle traveling at free-flow speed could advance in 5 seconds). Regional microsimulation approaches do not promise a unique, equilibrium solution, but rather a “stable” solution.

Generally speaking, analytical DTA solutions are thought to be more precise, but tend to have the most burdensome computational requirements. Heuristic solutions, which are more of an approximation to DTA, vary by sophistication, with the simplest ones being the least costly DTA approach, but also the least precise. Microsimulations are usually less computationally burdensome than analytical DTA but more so than most heuristic approaches. In addition, microsimulation approaches maintain a different theoretical basis, allowing multiple feasible and equally plausible network assignment outcomes.

Study objectives

The objectives of this study were to implement, test and evaluate alternative DTA programs with an eye towards possible deployment within ODOT or a partner agency, such as an MPO. We evaluated each program according to its analytical performance, computational performance, resource requirements, demands placed upon the user, information content of outputs, and any unexpected idiosyncratic issues that arose in the course of setting up and running DTA models. The overarching goal was to provide realistic expectations. It is our intention that the information contained in this report would: (1) help TPAU and other agencies to decide whether a DTA program is appropriate for the type of work they plan to undertake; (2) lead to a more informed program selection process; and (3) help to develop a realistic work plan for DTA program deployment and usage.

Remainder of this report

In the remainder of this report, we discuss the selection of the DTA programs considered in this study, providing a brief description of the background and relevant design features of each. This is followed by a section describing the set up of the case study, its input files, and database

management. Since an essential component of this work is to describe the process of model development, we spend the next section of the report discussing our staged approach to building, testing and troubleshooting the DTA networks. In doing so, we describe the common problems we encountered and their resolution. Next we discuss the results of testing scenarios we ran using each program, comparing the results to various validation measures. These scenarios include variations in demand loading, signal timing assumptions, and sensitivity testing of several input parameters. We conclude this section with a description of potential next steps for model calibration and refinement. Finally, we provide an evaluation of our experience of users of these two systems, commenting on our perceptions of each program's data file handling, user interface, error reporting, documentation, and developer support. We conclude this report by discussing lessons learned and consider issues that an agency interested in implementing a DTA program should consider.

Selection of Programs

Three programs were initially selected for the case studies based on data gathering and review of programs, which were developed under a previous Agreement 24620, Work Order 2. The review is described in the report dated May 27, 2009, "Review of Extant Dynamic Traffic Assignment Tools: Initial Findings" prepared by the Center for Urban and Regional Studies at Portland State University. The programs chosen for case studies were MATSIM, DynusT and Dynameq. These choices were made in consultation with TPAU staff, based on consistency with objectives for program architecture, performance features and cost to evaluate.

The three programs are somewhat different in their approach to DTA, but share at least two common methods. The first commonality is that all DTA programs solve the time-dependent-shortest path (TDSP) problem in which a theoretical traveler finds the least-cost (travel time) path through the network, given a specific initial departure time. This is described as "perceived" travel time because it is theoretically what a real-life user would experience. At each time interval (e.g., 6 seconds), travel times on the links are updated based on new traffic conditions for that interval.

The first commonality is time-dependent demand data are used as inputs. DynusT and Dynameq require this to be in the form of an origin-destination (OD) trip table specific to time intervals or a list of trips by departure time. Given the magnitude of demand loadings, developers of these programs recommend aggregating demand into intervals of 5 to 30 minutes, with 15 minute intervals being preferred. For example, a 3-hour assignment period with 15-minute demand intervals would require 12 OD trip tables. MATSIM differs due to the fact that its traffic simulator is actually part of a larger agent-based demand and supply system and therefore requires that individual trip plans with specific departure times be created as inputs.

One of the fundamental ways in which these three programs differ is in the flow models they use to propagate traffic through the network and derive travel times, representation of traffic signal operations, and the ways in which the programs converge upon a solution.

Both DynusT and Dynameq aim to provide a dynamic user equilibrium solution through iterative assignment steps. At each iteration, the programs calculate the TDSP, based on travel speeds determined by a traffic flow model. This process iterates until the outcomes of these TDSP calculations achieve an acceptable "relative gap." The user may specify a convergence tolerance expressed as percent relative gap or may specify a maximum number of iterations. The condition for dynamic user equilibrium is that, for a given departure, all used paths that connect an OD pair have the same travel time, and no user can switch paths and improve their travel time. The relative gap is the average proportional difference between the lowest travel time path and competing paths. At an absolute equilibrium point, the relative gap would be expected to be zero percent. In practice relative gaps of 2 to 5 percent are considered acceptable for DTA.

MATSIM aims to achieve a stopping condition analogous to dynamic user equilibrium through what is described as "systematic relaxation." In its full implementation, this means not only changing routes, but potentially entire trip plans (routes, locations, modes, starting times), although it is possible to restrict the relaxation options to route choice. At each iteration, agents'

day-long activity-travel plans are assigned to the traffic simulator and scored using a utility formula. Using a re-planning algorithm, which incorporates elements of learning and rule-based decision making, a select subset of agents modify some dimensions of their plans in order to obtain a better utility score on the next iteration. This continues until some percentage of agents can no longer find a better set of plans or until the maximum number of iterations is reached. If the modification options are restricted to modifying only routes, then the traffic simulator should behave roughly similar to other DTA programs, although this requires custom programming.

Each of these programs is described in more detail below.

DynusT

Background

DynusT was developed primarily by Dr. Yi-Chang Chiu of the University of Arizona, but has common roots with the Dynasmart-P program developed by Dr. Hani Mahmassani, now of Northwestern University. In addition, DynusT utilizes a graphic user interface (GUI) called NEXTA that was developed by Dr. Xuesong Zhou of the University of Utah. NEXTA was developed under contract to the Federal Highway Administration (FHWA) as an interface for TRANSIMS, as well. DynusT is distributed free of charge and was just released as an open-source program in 2011. In the past, FHWA has facilitated the deployment of DynusT as well as TRANSIMS through sponsored research and technical outreach efforts. The integrated GUI enables network editing and setting parameter values as well as graphic and tabular visualization of model results and diagnostic tools.

Design Features

DynusT is designed to provide an equilibrium solution for the DTA problem through simulation. Traffic flow is propagated through a "stimulus-response" type of algorithm described as anisotropic mesoscopic simulation (AMS), in which the rate at which vehicles move forward is governed by the density of vehicles within a downstream region of influence (e.g., 200 feet). Densities are recalculated at 6-second simulation intervals. Speed is derived from the well-known speed-density curve relationship, the parameters of which may be defined by the user, including saturation/service flow rates, jam densities, gap acceptance for permitted left-turns across oncoming traffic, and passenger-car-equivalent (PCE) values for trucks.

PCEs are used to adjust the physical capacity of links, but only on when going uphill and only on freeway facilities. Arterial throughput remains based on a unitary vehicle. The PCE for trucks is 1.5 by default for grades less than 2 percent, following Highway Capacity Manual 2000. To take advantage of this behavior requires that links be coded with percent grade values. Currently, DynusT does not permit multiple PCE values for more than one truck type.

As currently implemented, DynusT can represent fixed-time signal plans with and without coordination, or fully actuated signal timing plans without coordination. It cannot represent semi-actuated timing plans or actuated-coordinated timing plans. It may also be run without signal timing plans, in which case default traffic movement rules take over and vehicles take

turns going through intersections. It also does not allow the user to specify multiple timing plans for different periods of the day within the same DTA simulation, making it impossible to run DTA simulations that span multiple time periods during which signal timing plans change. The developers of DynusT are planning to add this capability in future releases.

One salient feature of DynusT is the ability to save paths from previous runs. These saved paths may be used in subsequent analysis in several ways. First, the user may wish to examine the paths through a select link analysis. Second, the user may want to use a set of paths derived under baseline conditions as a starting point for a single-shot estimation ("one off analysis") of a policy scenario involving non-recurring congestion, such as incident or event-management. A third use might also be to save a portion of system users and "freeze" their paths under the assumption that they have formed habits that will not change in response to facility changes, while allowing the remaining portion of demand to find a new equilibrium point.

Another salient feature of DynusT is a separate calibration program, which is currently not part of the main module. This module "wraps around" the main DTA program. It works by comparing assigned volumes to traffic counts and iteratively shifts the timing of demand loadings such that the resulting flows more closely match the counts. The idea is that the initial time-dependent demand tables are probably not reflective of actual demand in localized areas and that using local counts to shift the demand in time is a reasonable way to more accurately reflect actual diurnal patterns.

Dynameq

Background

Dynameq is a commercially licensed product of INRO of Montreal, Canada, and one of a small suite of travel demand modeling products offered by the company, which include the well-known EMME/2 and 3 products. It has an integrated GUI that enables network editing and setting parameter values as well as graphic and tabular visualization of model results and diagnostic tools. The licensing fee is tiered and set as a function of the network size. Currently, the maximum problem size that Dynameq can handle is limited to 10,000 links, 5,000 nodes and 1000 zones; however, INRO expects to increase problem-size limits in future releases.

Design Features

Dynameq is designed to provide a user-equilibrium solution to the DTA problem through iterative event-based microsimulation of traffic flows. The traffic flow model in Dynameq is based on car-following behavior and includes micro-scale gap acceptance and lane-changing behavior. This event-based design leads to travel time updates triggered by changes to network conditions, which in most cases will be a more computationally efficient method than updates at regular intervals. Consequently, definitions of vehicle lengths, lane dimensions and connectivity are important inputs. Multiple truck types may be defined with varying PCE values. Speed-density curves with user-modifiable parameters, including saturation/service flow rates and jam densities, are used to calculate travel times. The simulation uses an incremental approach to loading the network, requiring at least ten

iterations, after which the program switches to an equilibration mode. Dynameq uses the Method of Successive Averages (MSA) for this process, and the user can choose between regular or flow-balancing MSA (flow-balancing is the default).

Dynameq also has a couple of other algorithms that may prove useful in producing more efficient assignment runs. The “path pruning” option sets a path’s flows to zero if it drops below a predefined value as a fraction of total demand for an origin-destination pair and redistributes it proportionally to other paths. The “dynamic path search” option looks for new shortest paths to add to the past set during the second part of the DTA, when normally no new paths are being added, to replace paths with zero flow. If dynamic path search is activated then it is possible for paths pruned during path pruning to be added back into the assignment set in later iterations.

As currently implemented, Dynameq can represent fixed time signal plans and coordinated signal timing. It cannot represent actuated controllers. It is also possible to run a Dynameq simulation without signal timing, in which case default traffic movement rules take over and vehicles take turns going through intersections, lower volume roads yielding to larger-volume facilities. In addition, it allows the user to input distinct signal timing plans for different time periods within the same DTA simulation, making it possible to run a simulation that spans multiple hours of the day or even a 24-hour assignment, during which signal timing plans change.

Dynameq’s network performance diagnostic tools are quite similar in functionality to those of DynusT, including the ability to save paths and perform select-link analysis. INRO is planning to add a calibration tool in the near future and distribute it with future releases of Dynameq. This tool has been developed for internal use in diagnosing and fixing problems related to finding bottlenecks in the network and tracing the propagation of backups to specific capacity problems.

MATSIM

Background

MATSIM is distributed free of charge under GNU public licensing agreements by its developers through ETH-Zurich and TU-Berlin. There are actually two similar, alternative DTA modules in MATSIM, both based on event-driven queue-based model of traffic flow, developed to minimize computational time while providing an acceptable level of mesoscopic behavioral resolution. The traffic flow model is parameterized by in-flow and out-flow service rates on links, minimum gap requirements for in- and out-flow behavior, and maximum link capacities.

Design Features

As MATSIM was designed for application on very large-scale problem sizes, such as the Swiss national travel model, it leaves out certain details found in other DTA programs, such traffic control timing plans, intersection geometries, and turn movement priorities. The program does not model signal timing plans and phasing directly. Rather, it assumes an average fraction of green time for each intersection movement, which is modulated up and

down during the course of the simulation in response to demand, much like an actuated system. This would seem to make it more appropriate for use at national, state or regional levels of analysis and less appropriate for studying design and operational improvements or work zones, and may provide insufficient resolution for focus areas and corridor studies.

Nevertheless, because MATSIM is free and known to be computationally efficient, it was an attractive alternative. One of its attractive features is a visualization module, called OTFV, which allows the user to view pseudo traffic flows on the network. On the down side, MATSIM does not have a GUI for network editing and entry of other input data. All inputs files must be created by the user in XML format, and the program itself is contained in JAVA language JAR files.

Implementation Problems

Once we acquired MATSIM and attempted to implement a version of our study area network, problems arose that caused us to reconsider its use in this study. Foremost was that our objective in this study was to use the DTA module of MATSIM separately from the main portion of the program. This was of particular interest because we assumed that, in Oregon, DTA would most likely be used to supplement the trip-based models currently in use. We were optimistic about these prospects since a research team from University of Toronto had recently used the traffic simulation portion of MATSIM and integrated it with a different activity-based modeling system (Hao et al., 2010).

We found that the activity-based demand part of MATSIM was integrated with the DTA portion. To use MATSIM as we had intended would require a significant amount of programming in Java. We learned that the Toronto team sent a graduate student to Berlin for one month to work directly with the developers, but this level of involvement was beyond the scope and budget of our study. Attempts to "trick" MATSIM by recasting our demand tables as MATSIM plan files did not work and was unduly cumbersome, despite encouragement from some of its developers. Faced with a potentially lengthy and risky programming effort that was outside the scope of this project, we decided to go no further with the evaluation of MATSIM, after consultation with and approval by the TPAU project manager.

Additional factors that contributed to this decision are worth mentioning. First, there is the lack of detailed model control over the representation of signalized intersections, which would be desirable for some studies. The lack of detail on intersection control would seem to make MATSIM most appropriate for analysis at a statewide or regional level as it is currently used in Switzerland. Second, MATSIM currently lacks most of the network diagnostic tools found in DynusT and Dynameq for analyzing network performance, including select link analysis.

Finally, as a university-based research project, MATSIM is constantly evolving, which will undoubtedly lead to innovation and an improved tool, but this also means that user support is limited. There is a growing user community and now annual user-group conferences in Europe; however, this rather informal support network is a concern.

Case Study Setup

Study area

The study area chosen for this project encompasses a portion of the Portland regional modeling system and is shown below in Figure 1. The study area is centered on the City of Beaverton and includes adjacent portions of other municipalities, rural Washington County, and the western edge of the City of Portland in Multnomah County. The transportation network includes heavily traveled sections of U.S. 26, running east-west along the northern edge of the study area, and OR 217, running north-south along the eastern edge of the study area. These two limited-access freeways have an interchange in the northeast corner of the study area. Beaverton has a fairly dense retail-commercial center, and Nike Corporation's corporate headquarters is located near the western edge of the study area. The area is suburban in character and is served by one light-rail line, one commuter rail line, and an extensive bus network. It was chosen by Metro travel demand modeling staff members, who believed this area would make a good case study due to the dense Beaverton commercial core and the confluence of U.S. 26 and OR 217. OR 217 has also been the subject of other recent studies on potential traffic flow improvements, focusing on the tight spacing of its interchanges and the problems this poses for traffic flow.

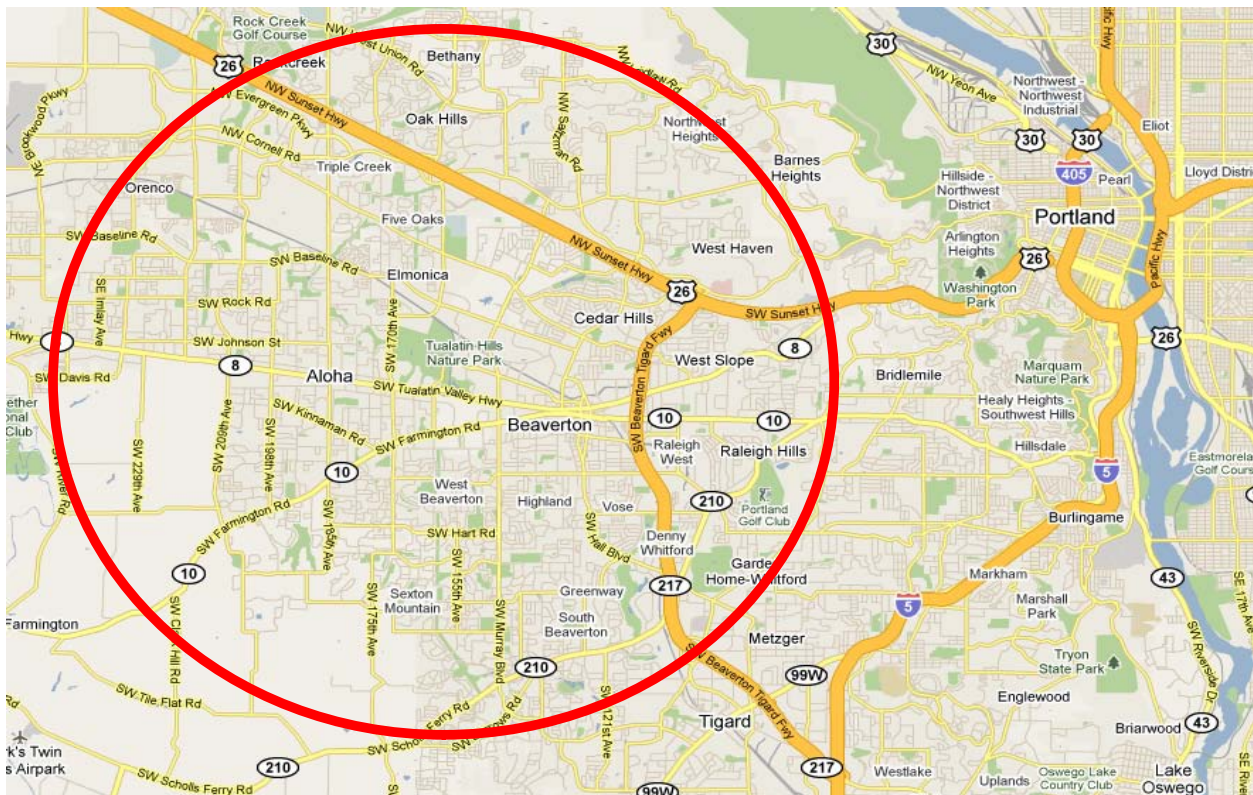


Figure 1. Study area

Network files

Metro created the study area as a sub-area network from its regional modeling system and provided the demand for the study in the form of trip tables. The network itself represents 2009-2010 conditions, while the demand is based on an estimate for a 2005 model year. Due to recessionary effects on economic activity and travel since 2005, it is expected that the 2005 demand estimates may be even higher than what would be observed for 2009. In total, the network includes 106 traffic analysis zones (TAZ), 1,560 links and 630 nodes. The study area also includes 135 signalized intersections. The subarea was extracted from the regional network, and the assigned volumes at each of the nodes corresponding to an external station in the study area became the demand for that station for that time period. Figure 2, below, shows the extracted subarea of the regional network in VISUM 11.02. The network links and nodes were originally created by Metro from NAVTEQ files, due to their centerline accuracy, with modeling attributes added from an existing highway network model.

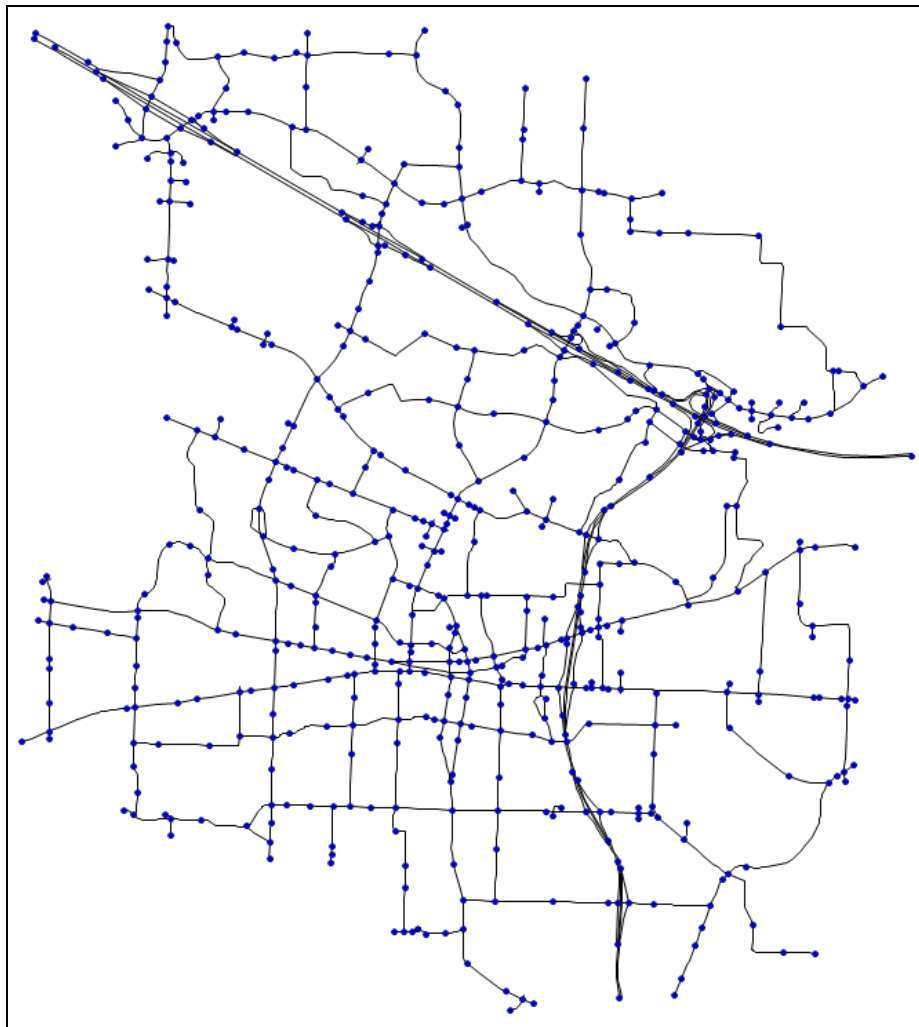


Figure 2. Extracted subarea of Metro regional model in VISUM 11.02

Demand data

In order to generate demand for these external stations, Metro first ran separate, static network assignments on the full regional network in VISUM for eight time periods that cover the full 24-hour average week day:

- AM 1-hr shoulder (6 to 7 a.m.)
- AM 2-hr peak (7 to 9 a.m.)
- MD 5-hr Midday (9 a.m. to 2 p.m.)
- PM 1-hr shoulder "A" (2 to 3 p.m.)
- PM 3-hr peak (3 to 6 p.m.)
- PM 1-hr shoulder "B" (6 to 7 p.m.)
- EV 3-hr evening (7 to 10 p.m.)
- NT 8-hr night (10 p.m. to 6 a.m.)

For each time period, there were four demand segments:

- Single-occupancy passenger vehicles (SOV)
- High-occupancy passenger vehicles (HOV)
- Heavy trucks
- Medium trucks

Nodes representing the edges of the network became external stations for the DTA network study area. External stations were redefined as TAZs and their demand incorporated into the full trip table along with the OD flows generated by internal TAZs. Metro created these demand tables along with a diurnal distribution of trip starting times by time of day from the 1994 Metro household survey, shown below in Figure 3. This single distribution aggregated all passenger trips of all types and did not include medium or heavy truck traffic. In the absence of more current information, this would seem to be a good estimator of general diurnal travel patterns for the region. We used this diurnal distribution to allocate demand to 15-minute intervals within each of the eight time periods listed above and this was applied to all four demand segments.

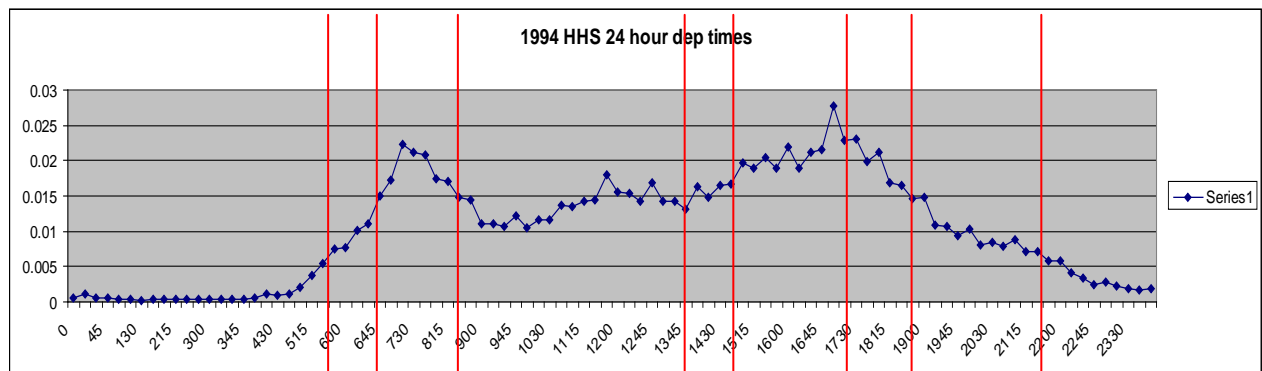


Figure 3. Distribution of person travel by time of day from Metro 1994 household survey: Vertical red lines represent the boundaries of the eight demand matrices provided by Metro.

Common database

Our approach to network development was motivated by the requirement of creating DTA network models that would be as equivalent as possible across multiple software platforms. To achieve this consistency, we created an independent database in MySQL as a repository for all network input data, including links, nodes, signal timing plan and phasing information, turning movement allowances, and demand tables. Using this common starting point, we wrote scripts in Python and SQL to use these database tables to create the input files needed by DynusT and Dynameq. As the work progressed, we found it necessary to make changes to the original input data in order to reconcile network coding errors; to input more realistic signal timing, phasing and movement data; and to provide additional detailed information that was not available in the original VISUM network. In addition, certain scenario runs required changes to either network attributes, traffic control plans, or demand loading. Since, in most cases, these changes were first made in the MySQL database, it was relatively easy to rerun the Python/SQL scripts to generate new sets of inputs for both programs and thereby ensure that the changes would be made consistently in each program.

Some network changes, however, were not so easily made in the database. For example, it was necessary in some places in the network to move, add or delete a connector link, and such changes were made manually using the network editors of both programs. In addition, in Dynameq the creation of turn bays required us to split links and add the correct number of lanes in each, which we did using the network editor GUI. We did not split links for DynusT since the presence and number of turn bays are simply attributes of a larger link and do not require separate link records. In addition to network editing changes, there were certain signal control parameters that were present in one program and not in the other, and these changes were made in the individual programs through their interfaces. Namely, in Dynameq the ability to prioritize certain permitted movements over others, using templates, and to prohibit right turns on red was applied in some instances in order to resolve problems at certain intersections.

DTA Network Model Development

The process of developing a dynamic network-based travel model requires an initial setup, followed by multiple test runs to learn about program performance, identify problems, make adjustments, and calibrate the model. Integral to this process is learning about and testing the various settings that can influence program performance and results.

In STA links can have V/C ratios greater than 1.0 and thereby carry greater volume than their capacity, but still produce useful results. The consequences of over predicting demand in a DTA are much greater. If more demand than the network can handle is loaded into a DTA network, queues build, blocking back intersections and ramp ingress and egress. This can quickly escalate to the point of widespread gridlock where new demand cannot enter the network and the network cannot empty out, in which case the model results are useless. This has two important implications for the way in which an agency handles analysis differently with DTA compared with STA. First, it is imperative that the demand loaded onto the network in the highest peak periods of the day be able to "clear" the network at the end of the simulation in a reasonable amount of time. Second, agencies that are accustomed to running future-year scenarios in which a future-year demand is loaded onto a current-year network for the purposes of identifying future-year deficiencies may be surprised to learn that they cannot perform the analysis at all, or at least not the same way as with STA.

Both of the DTA programs studied here provide detailed output information that allows one to review the performance of the network over the period of the simulation. Both DynusT and Dynameq provide an essential diagnostic tool in which the analyst can load a finished assignment run and, using a sliding lever, step through time intervals, setting performance indicators such as density, queuing and speeds, and watch how these performance indicators change on a network map during the study period. With experience the analyst soon learns how to spot where a bottleneck begins and how queuing propagates problems to other links and intersections. The usual practice is to run the DTA simulation for an extra time period (e.g., 1 or 2 simulated hours after demand is scheduled to load the network) and watch to see when the network clears, or if it fails to clear. If the demand cannot clear after an extra hour or so, then this is cause for making immediate adjustments.

Incremental approach

Consistent with recommendations for simulation network development by Dowling *et al.* (2004) in *Traffic Analysis Toolbox III: Guidelines for Applying Traffic Microsimulation Software*, we followed an incremental approach to development of the DTA network model. This process may be described in terms of four major stages in which we gradually added complexity and demand, at each step diagnosing and attempting to resolve problems related to network performance.

The objectives were simply to resolve network flow problems, not to perform any type of calibration. Such problems manifest themselves in excessive queues that build at various junction points in the network. Some problems are related to the ability of vehicles to enter and exit the network at TAZ connection points. Other problems result from intersection capacity problems in which queuing builds to the point of blocking back other intersections, freeway ramps, or even mainlines, and may lead to rapidly propagating gridlock that does not clear by the

end of the simulation. When such problems occur, the network is not performing realistically, and the problems must be diagnosed and resolved before meaningful analysis can occur.

It is worth mentioning that we performed most of the trouble-shooting and network and signal adjustments in Dynameq due to two factors. This came about because Dynameq is more sensitive to network details than DynusT. Once we were past some of the more obvious network and intersection control coding problems, additional problems would typically be first identified when running Dynameq. Second, we found the network editor in Dynameq to be easier to use; therefore, we would use it to make the changes, then export these changes to the MySQL database and use them to make the equivalent changes in DynusT.

Common sources of network flow problems include: incorrect link capacities, particularly for centroid connectors; poor placement of centroid connectors, particularly at the edges of the network and near major intersections; missing turn bays; incorrect lane geometries; incorrect representation of allowed movements at intersections; inaccurate representation of signal timing and phasing; and questionable demand at certain locations. The four stages of DTA network development that we followed were intended to approach the diagnosis and resolution of these problems in a systematic way. They are described below.

Stage 1. Basic network, default signal timing, partial PM Peak demand

Create basic network detail using default settings for signal timing and run with partial, flat (non-varying) demand loading for a single 3-hour period (PM Peak). Diagnose and troubleshoot network problems related to network connectivity, link capacities, centroid connector capacities and placement.

Consolidating links and nodes

Before exporting the network from VISUM, walk-to-transit links were removed. Unnecessary nodes reduce DTA travel times and run times; therefore, the remaining nodes were consolidated by combining links that share all attributes and removing nodes in between them. Figure 4, below, shows the difference in number of nodes in DynusT before (left) and after (right).

Centroid connectors

An important network design consideration is how to represent network loading points. In DynusT, this is accomplished using generation links, destination nodes and virtual links. In Dynameq, these are referred to as connector links, centroids and virtual links. Developers of both programs recommended eliminating as much as possible the centroid connector links inherited from the VISUM network. However, Metro initially expressed the desire to maintain existing centroid connectors for comparability and backward compatibility with their static assignment model. Moreover, we reasoned that, since the two programs treated these connections slightly differently, we were more likely to maintain comparable networks if we maintained the original centroid connector structure as much as possible. In later phases of network development and testing, we tried to make connectors more similar to local streets and driveways, with only one or two lanes, and added stop signs for connectors entering the network mid block.

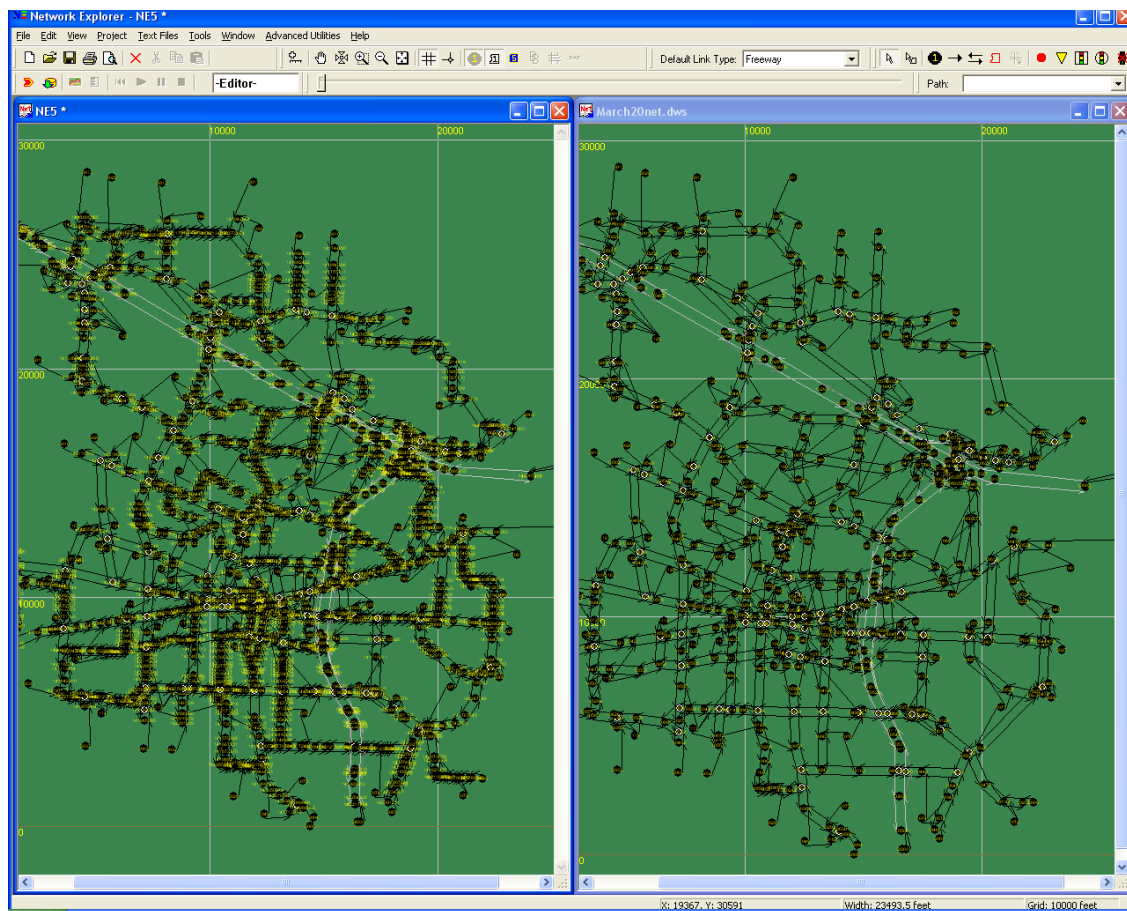


Figure 4. DynusT network before (left) and after (right) removal of unnecessary nodes

Unsignalized junctions

Dynameq uses intersection templates to set priorities for different intersection control types (all-way stop, two-way stop, and yield). The templates use road facility numbers to create a hierarchy of movements at two-way stops and yield/merge signs, where higher number facilities (typically lower volume roads) yield to or stop for the larger-volume facility. In DynusT, equivalent templates do not exist and these relationships need to be coded on a case-by-case basis, or by applying a rule provided by the user. Yield signs were coded for all freeway on-ramps.

Default signal timing

One reason to consider default signal timing plans is to determine to what extent an agency that wants to run a DTA program for a regional model needs to invest in the effort of coding realistic timing plans. The default settings for signal timing were coded by Metro into a VISUM project file, and we extracted these timing plans directly from the VISUM database. The default timing and phasing plans were generic across all signalized intersections and were coded as 25 seconds of green, 4 seconds of yellow, and 1 second of all red time for

every phase and approach. These default settings did not vary by time of day. Protected left-turn phases and other, more sophisticated timing and phasing plans were not represented in these default settings. Fixed-time signals were assumed, and all intersections were uncoordinated.

Turn bays

Turn bay representation is a basic network consideration. From VISUM we had information on number of turn bays and whether they represented left or right turns. In DynusT, the presence and number of turn bays are simply attributes of the link entering an intersection and do not require separate link records. In DynusT the lengths of turn bays take on a single global value (we used the default value of 200 feet).

In Dynameq, turn bays must be created explicitly, and this is accomplished most easily using tools in the GUI designed for this purpose. Links terminating at intersections are split in two, and lanes are added on the newly created link nearest the intersection. Links upstream and downstream from the turn bay lanes are then realigned with the through-lanes. Figure 5, below, show the outcome of the splitting and realignment of one intersection. Ideally, the length of the turn bay link should match reality and each turn bay could be adjusted using the manual tool with comparisons to aerial photos and GIS or Google Maps. For this basic effort, we reasoned that the effort of manually adjusting each turn bay was beyond the scope of this study. In making a few example comparisons, we found that in almost all cases the lengths of turn bays which we created in an arbitrary fashion were longer than their actual lengths, which we felt was the “right” direction of error, since capacity was at least not being reduced relative to field values.

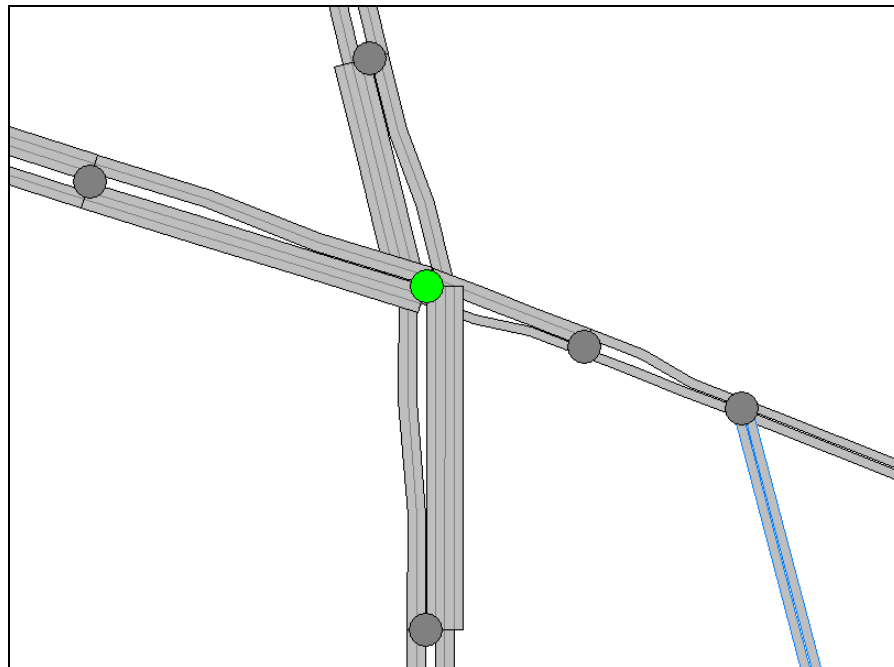


Figure 5. Link editing in Dynameq: turn bays are considered separate links with a higher number of lanes.

Warm-up and cool-down demand periods

For a given time period of analysis, best practices for DTA (Chiu *et al* 2010) recommend a warm-up period of about one hour in order to load the network for the beginning of the analysis period. It is also recommended that the simulation be run for at least one hour after the analysis period, with additional demand in order to create realistic expectations on the part of drivers as to their travel times. The idea here is that drivers beginning their trips at the end of the study period will make path choices based on the total path time which, if their trip ends in the subsequent time period, should reflect congested travel times in that latter period. Moreover, additional time periods are run without demand in order to allow the network to clear, or to determine that it will not clear. Thus, in this particular case, we established a study period of 3 to 6 p.m. There was a warm-up period of demand loading from 2 to 3 p.m., and a cool-down period of demand loading from 6 to 7 p.m., followed by an additional two hours of simulation without additional demand, which we monitored for network clearance.

Problems with centroid connector loading

We first tested network performance using only the SOV demand (72 percent of total daily demand). Even with this reduced demand, we found numerous bottlenecks. Since at this stage, we were using default timing plans, we temporarily ignored problems at signalized intersections and focused on problems related to bottlenecks at external stations and other TAZ connectors. The most common problem found at this stage was inappropriate capacities for centroid connectors. Specifically, we found that the centroid and external station connectors imported from the VISUM network were coded with unrealistically large capacities (9 lanes) and free-flow speeds of 12 mph, a vestige of the trip-based model practice of providing unlimited capacity for loading the network and lower speeds to reflect local street conditions. In both DTA programs, the huge capacities resulted in a problem of loading the network too quickly, with jams forming at loading points. Following the recommendations of both program developers, we recoded connector links to have the same number of lanes as the arterial links to which they connected, and this provided the smoothest loading. We found that using fewer lanes usually resulted in vehicles queuing and not making it onto the network. In addition, we found it necessary to increase speeds on the connector links to 20 mph, as recommended by the developers.

DTA developers recommend against transferring the centroids and connectors directly from the STA model, as these are typically abstractions. In a DTA network all connectors should represent real-world driveways. Metro had already enhanced the connector placement using aerial photos, and it was outside the scope of the project to adjust the number and location of connectors. Retaining the connectors from the STA model caused easily detectable problems in a few locations, and we adjusted connectors in such cases as needed. In some cases, the physical placement of a connector link was a problem, particularly at the edges of the network, leading to and from an external TAZ. A few external TAZs had multiple connectors to the network, but it was clear from running the programs that the resultant demand loadings were favoring one particular connector over the others, or one was not being used much, if at all. In a handful of cases, we either moved or deleted a connector link to provide a more balanced distribution of flows entering and exiting the network at a particular TAZ.

The physical distance between an intersection and a connector entering mid block was also a prevalent problem. This differs from static network assignments because, in a DTA network, the physical distance between intersections matters due to queue lengths. An example is shown below in Figure 6 in which a connector loads very close to a major intersection. Moving the problematic connector to another, nearby location in some cases caused new congestion problems in the new location.

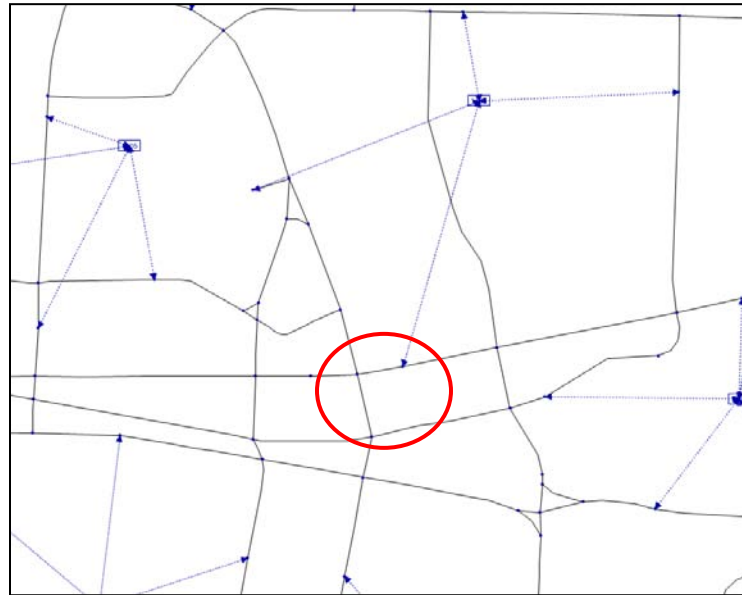


Figure 6. A centroid connector placed too close to a major intersection.

Performance of default signal timing

It quickly became apparent that the default signal timing plans were problematic, even at this relatively low level of demand. With DynusT, one can avoid having to use realistic timing plans by using default timing plans that assume fully actuated signals throughout the study area and adapt to varying demand patterns. The user supplies minimum and maximum green times for each movement; however, signal coordination between intersections is not available for actuated signals. With Dynameq, signal actuation is not currently available, making realistic fixed timing plans critical under any significant level of demand. For the sake of comparability we ran both programs with fixed signal timing plans.

A third option is to run the DTA simulations in either program without signal timing plans, in which case the behavior in both programs is to have vehicles take turns, similar to what happens in real-life when a traffic signal stops working and it reverts to a blinking red phase for each approach. This is unrealistic, but we found that programs run this way were able to accommodate more demand than under the more realistic signalized scenarios. The reason for this is that intersection throughput becomes a function of demand at each approach and movement and is not constrained by green time settings. Running either Dynameq or DynusT

without signalization seems to be tantamount to fully-actuated control without minimum or maximum green times. This would seem to be the easiest way to determine whether the network itself, independent of timing plans, can accommodate the demand fed into it. We ran both simulations without signal timing plans and compared the resulting assigned hourly volumes to traffic counts to help us determine whether the demand loadings were realistic. These scenarios are described below in more detail.

By the end of this stage, both networks cleared with no signal timing plans and partial, flat SOV demand. With default signal timing, DynusT cleared but Dynameq did not. Problems were particularly apparent near edges of the network where heavy demand was loading.

Stage 2. Add realistic signal timing plans

Add realistic signal timing plans and run with partial, flat demand loading for a single time period (PM Peak). Diagnose and troubleshoot network problems related to turning movements and intersection capacities.

Importing and interpreting signal timing plans

During this phase we obtained 24-hour weekday signal timing plans for the study area's 135 signal controllers. Three jurisdictions operate and maintain signal controllers in the study area, and we obtained data from the City of Beaverton, ODOT Region 1, and from Washington County through the consulting firm DKS Associates. Converting timing plans from signal controller format to formats that could be used by both DTA programs was a considerable challenge, an effort described below in Appendix A. To summarize, this involved interpreting timing plans from different signal timing manufacturers and translating their parameters in ways that could be used by both DynusT and Dynameq, which required several simplifications. Neither program is set up to handle sophisticated intersection timing plans, such as dual-ring controllers with semi-actuated phases and coordination between intersections. Both programs can handle fixed-time plans and coordination with other intersections using offsets. DynusT can represent fully-actuated signals, but not semi-actuated or coordinated-actuated. To provide a comparison among equals, we specified signal timing plans in both programs as fixed time, with offsets for coordination where available.

It is important to note that ramp metering exists in the study corridor on freeway ramps at certain locations in the AM and PM peak periods. For simplicity, we did not obtain and implement ramp meter timing plans, although both programs support their use.

Once an initial set of signal timing plans were developed for each program for the PM Peak analysis period, we ran the both DynusT and Dynameq using the SOV demand table. This revealed numerous problems related to turning movements and intersection capacity issues, as evident by jammed intersections, excessive queuing at certain approaches near the edges of the network, and counter-intuitive turning movements.

Whether to adjust timing plans

To consider adjustments to timing plans is a slippery slope due to the possibility in some locations of throwing coordinated signals out of synch; therefore, we avoided signal timing plan adjustments, especially in coordinated corridors. In addition, the developers of both programs advised against changing real-world signal timing plans and in favor of adjusting demand loading. Near the edges of the network, however, we often found that real signal timing plans gave insufficient time for traffic coming to and from external zones. Since these edge conditions are somewhat unrealistic to begin with and without an obvious basis for reducing demand, we decided instead to adjust signal timing capacities. An example of this is shown in Figure 7, below. Signal timing plans were modified to give more green time to the approach from the external zone.

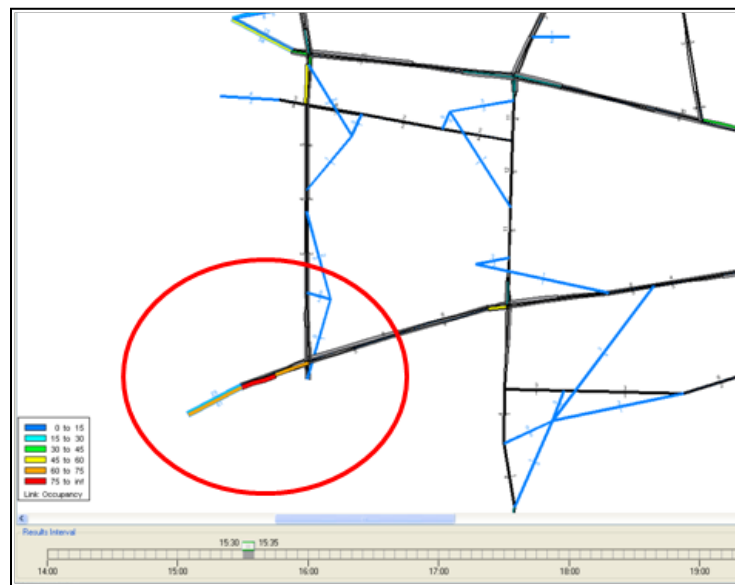


Figure 7. Heavy external demand loading caused improbable congestion at this intersection.

In such cases, the spatial distribution of traffic was likely to be skewed due to the sub-area extraction process. The remedy was usually to re-allocate green times, while maintaining cycle lengths, or to create a protected left-turn phase where one previously did not exist. In a few cases, we re-coded a movement as protected that was originally coded as permitted when we found that there were no conflicting through movements of vehicles and that the "permitted" status was due to pedestrian prioritization from imported Synchro files. Also, in Dynameq, one has the ability to allow or deny right turns on red. At the advice of INRO, for a couple of intersections we prohibited right turns on red that were aggressively impeding the flow of an opposing movement with green time.

Resolving turning movement problems

Both programs have built in default rules for allowed turn movements, but these need to be recoded manually in several cases. Another common problem was inconsistency between the turning movements allowed in the VISUM network and what was actually allowed by the signal timing plans. We resolved these issues through visual inspection of aerial photos of each intersection. In conjunction with the signal movement inspection we also verified turn bays. We found several discrepancies in turn bay numbers and direction (left/right) between the VISUM model and the aerial photos, and adjusted to the aerial photos. A classic example is portrayed in Figure 8, below. Westbound right turns are permitted at this intersection through a separate channelized lane, represented as a separate link in the DTA networks. This right-turn channel is not regulated by the signal timing at the intersection; rather, it yields. The initial coding of turning movements permitted westbound right turns at the intersection, but these would have been mistakenly made from through-lanes and were thus recoded as prohibited.

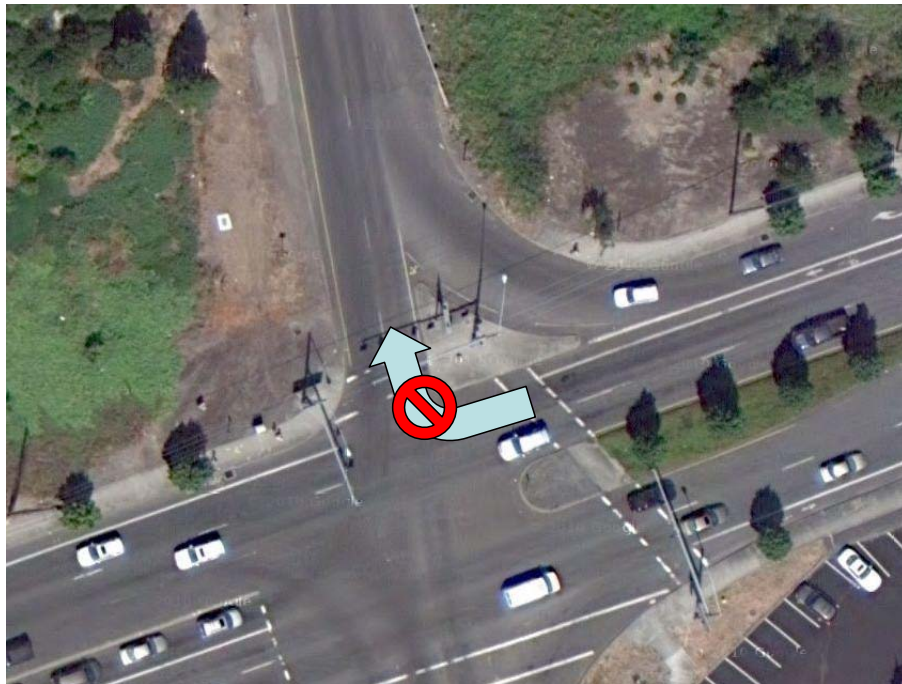


Figure 8. Initial turning movement coding allowed westbound right turns from the through lane.

In the original static model network, there are some instances where demand loads onto links with insufficient capacity, causing false congestion. At certain external stations, we had to relocate centroids and connectors to solve bottlenecks. Figure 9, below, illustrates one example in which study area boundaries resulted in an external station at Hall Boulevard, a major thoroughfare. The external station, TAZ 3005 was originally coded to load at a small cul-de-sac stub link on the north side of Greenway Boulevard. The demand modeled at this location is undoubtedly much higher than in reality, with hourly productions of 311 trips and hourly attractions of 785 trips.

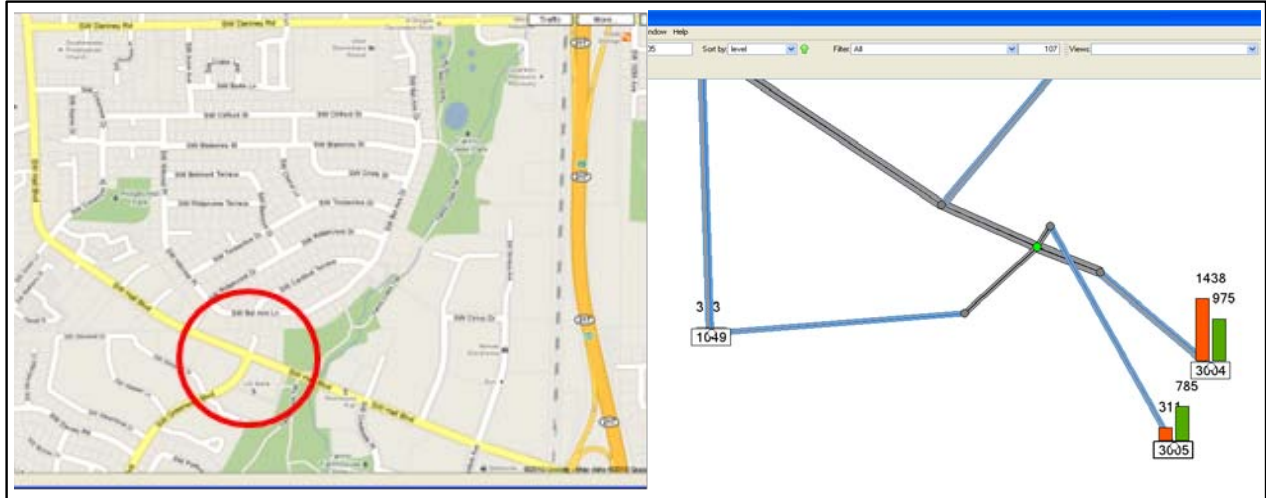


Figure 9. SW Hall Blvd & SW Greenway Boulevard location map (left) and corresponding DTA network demand loading (right) as an external station.

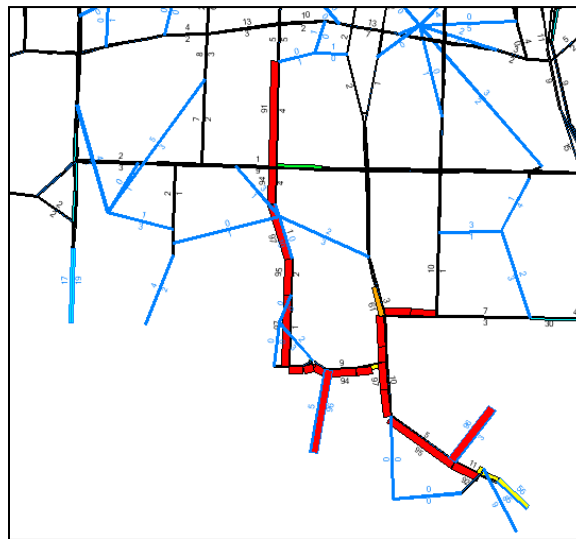


Figure 10. Resulting traffic density from initial placement of external station at a cul de sac.

The signal timing plan for this intersection allotted only 20 seconds per cycle to the eastbound left-turn onto this street, causing heavy congestion on the eastbound Hall Boulevard lanes carrying traffic trying to turn left to get to this final destination. The resulting congestion in the form of vehicle density may be seen in the Dynameq screenshot shown in Figure 10. Hall Boulevard's eastbound lanes had high occupancy as early as 15 minutes into the study period and proceeded to back up and block vehicles on Ridgcrest Drive trying to get onto Hall Boulevard eastbound. At the same time, TAZ/centroid 1049 on the other side of Hall Boulevard had almost no productions and attractions. The solution was to move centroid 3005 and its connector link across the road to the south side of Hall Boulevard, to the Greenway Blvd link. This resolved the congestion problem in this area.

By the end of this phase of development, with partial, flat SOV demand and realistic fixed-time signal timing, both programs cleared. We now considered the network ready for full demand, and expected to detect and resolve assignment problems.

Stage 3. Run with full demand, apply peaking factors

Run full demand, flat and peak-adjusted, with realistic signal timing plans for the most congested time period (PM Peak). Determine whether demand is over or under predicted for the network capacity by comparing to observed counts. Diagnose and troubleshoot additional problems.

Adding other vehicle classes

During this phase of network development we attempted to load the full demand onto both DynusT and Dynameq networks for the PM Peak period. This meant creating time-interval-specific demand tables/input files for SOV, HOV, medium trucks and heavy trucks. DynusT currently allows only one truck type; therefore, we combined medium and heavy trucks into a single demand segment. For Dynameq, we maintained separate segments with distinct passenger-car equivalent (PCE) values. For DynusT, the PCE for trucks was 1.5, whereas for Dynameq we used PCEs of 1.2 and 2.0 for medium and heavy trucks, respectively.

Given that SOV represents about 72 percent of all vehicle trips, in this stage we were loading 39 percent more vehicles onto both networks, not taking into account PCE impacts. According to the developers, DynusT only uses PCE values to effect the capacity calculations on uphill freeway segments, following methods found in the 2010 Highway Capacity Manual. This also implies that the network must reflect elevation changes, which would have required a substantial coding effort beyond the scope of this study. In Dynameq, PCE values affect the amount of space they consume in traffic, regardless of facility type or elevation, and thus have a more pervasive effect on the simulation.

The initial results of the 100 percent, flat demand loading were widespread gridlock situations in both programs. The following problem sources were plausible:

- Additional network coding errors
- Signal problems caused by unrealistic signal timing
- Assignment problems (too much demand loading at certain times).

Applying peaking factors

Using Metro's diurnal 15-minute distribution from the 1994 household survey (see Figure 3 above) we created new peak-adjusted demand tables. With peak factors, simulated volumes matched the daily traffic patterns better, but widespread gridlock still formed.

Attempts to resolve gridlock and accommodate demand

The screenshots in Figure 11 show the DTA run with 70 percent of total demand, on the left, and 75 percent of total demand, on the right. This illustrates how sensitive the programs can be to small increases in demand. Prior to this point in the study, connector links entered the network freely, even mid block. To ensure less interrupted flows on the network, we added stop signs where connector links enter the network, which helped resolve several bottlenecks.

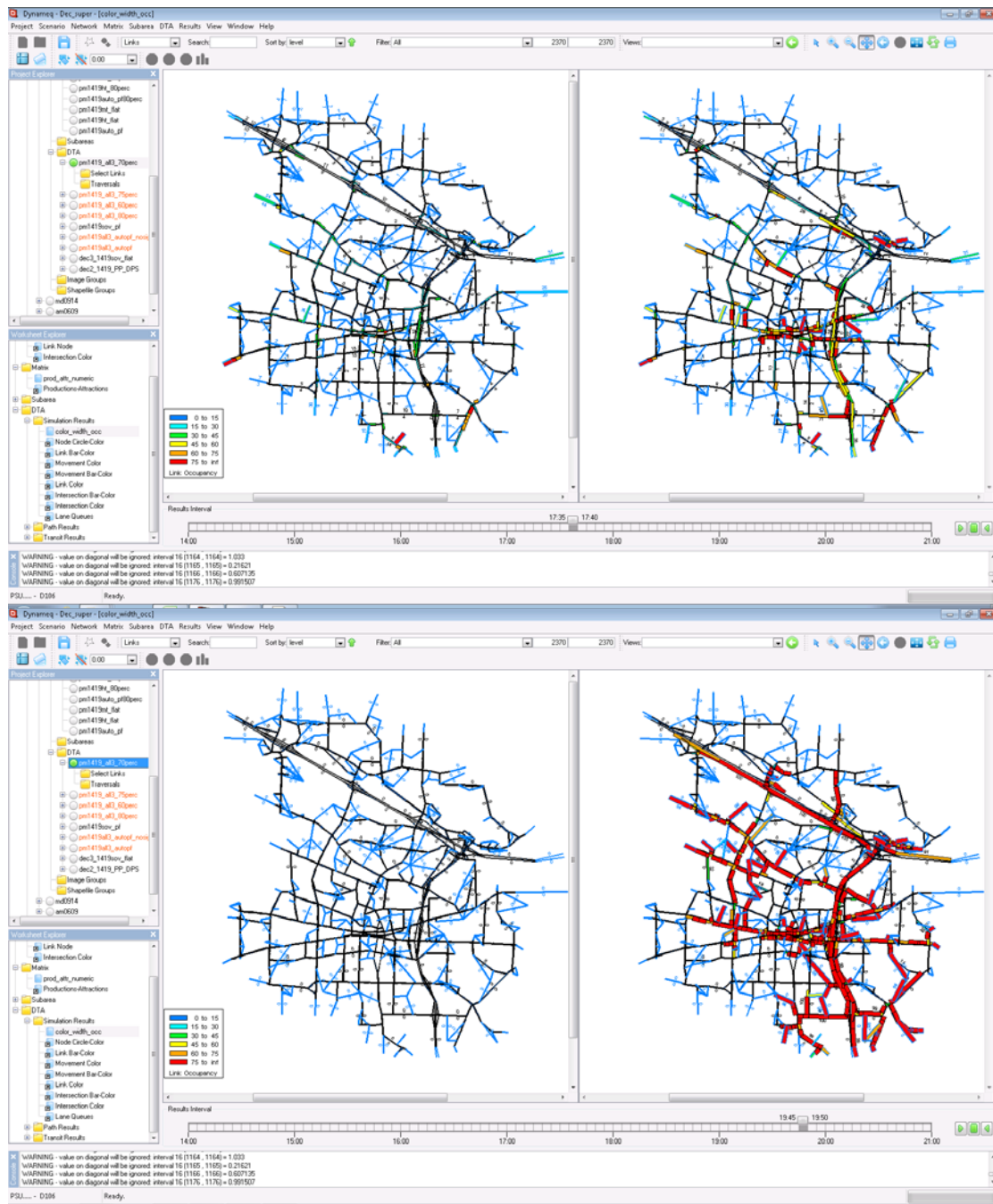


Figure 11. PM Peak network simulations with demand at 70% (left) and 75% (right): Top shows 3:40 into the study period and bottom shows 50 minutes after demand has stopped.

In DynaMiq and DynusT, we solved network coding problems at several locations where gridlock originated, but were unable to accommodate more than 75 percent of the total demand of the PM Peak period (75% auto, 75% medium truck, 75% heavy truck). As an additional check, we ran a 100 percent demand PM Peak assignment without signalization. This unsignalized run cleared in both programs. Comparing the sum of field counts to the

sum of simulated volumes on 86 count locations, we concluded that the model demand represented observed total volumes and was not unrealistically high.

It is not obvious what problems were limiting the capacity of the signalized PM network to 75 percent. In particular, though, problems emanated from a quadrant of streets centered in downtown Beaverton, an area bounded by SW Farmington Road, SW Cedar Hills Boulevard, SW Canyon Road and SW Hocken Avenue. This area, shown below in Figure 12, would be the starting point for queuing that would block back intersections and propagate gridlock in many of the scenarios.

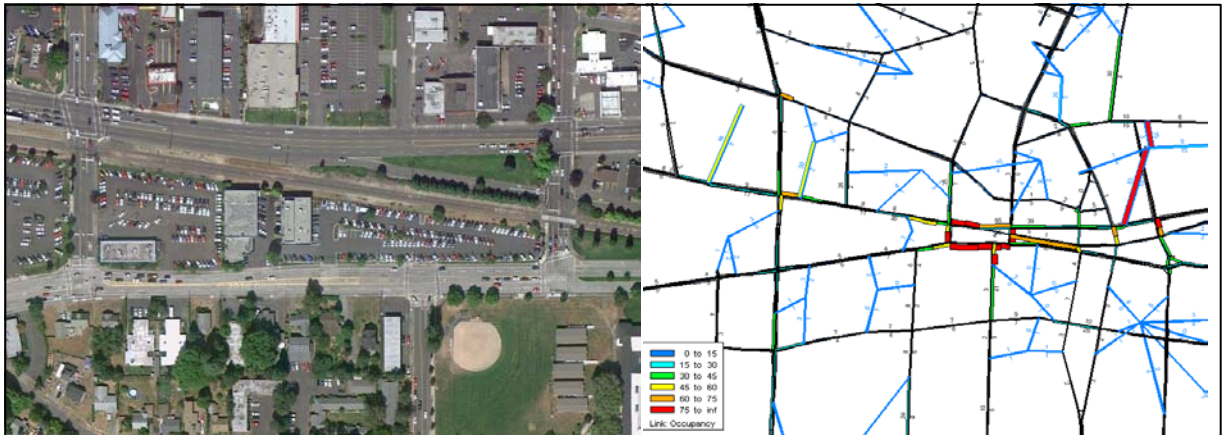


Figure 12. “Epicenter of gridlock” in Beaverton

Without deeper investigation, we conjecture that this tightly spaced arrangement of intersections and synchronized timing plans with offsets creates a situation in which there is little room to accommodate error under high demand. In addition, the fixed time signal timing plans that we implemented in both DTA programs do not provide the flexibility of the semi-actuated plans actually implemented in the field. It is likely that the actual demand patterns vary from the modeled demand patterns, at least enough to require some adjustments. To accommodate full demand and relieve congestion in this central quadrant, both DTA providers recommended calibration of the demand matrices, above and beyond applying peak factors. To do this we would have needed to obtain turning movement counts and additional link counts, preferably by time of day, in order to fully understand the variation in flow patterns.

Stage 4. Run AM Peak, Midday, and 24-hour scenarios

Run full demand with realistic signal timing plans for other time periods. Diagnose and troubleshoot additional problems.

AM Peak

After clearing network coding problems and accommodating traffic flows at critical intersections, we were able to run 100 percent of the demand for the AM period. The hourly demand of the AM period was 51,474 vehicles per hour, lower than the 59,006 hourly demand of the PM Peak.

Midday

The Midday period (MD) also ran well at 100 percent of the demand in both programs, and we decided to use the Midday for sensitivity testing. In light of our experience with the congested PM Peak, we felt that the somewhat lower hourly demand in the Midday period compared with the AM Peak would give us more of a buffer for sensitivity testing and running saturated demand scenarios.

24-hour run

After working on individual demand periods until they cleared, (except for the 100 percent PM which never cleared signalized) we ran all demand tables consecutively in a 24-hour run. For a given intersection location, timing plans may switch between peak and off-peak versions at different times of day, and in some locations there may be as many as four different plans. Since the exact timing of the switching differs by location, we generalized the process by grouping the signal timing plans into the following four time periods:

- AM = 06:30 – 09:30
- OP1 = 09:30 – 15:00
- PM = 15:00 – 20:00
- OP2 = 20:00 – 06:30

The 24-hour scenario could only be run for both programs without signals, and only at 75 percent of demand. We tried 100 and 75 percent demand with signalization in Dynameq, but neither cleared. Since use of multiple signal timing plans was not enabled in the DynusT GUI, we could not perform the same the test. Additionally, we ran the 24-hour scenario without signalization and 100 percent of demand. This scenario cleared in DynusT, but not in Dynameq. A run with 75 percent demand and no signalization cleared in both programs.

An important lesson was that, even when individual demand periods, such as the AM and Midday, cleared when run separately, this did not hold when the two time periods were run consecutively. Despite having run a 1-hour warm-up period for the Midday scenario, during the 24-hour run, congested conditions were created in the AM peak that resulted in a different starting demand pattern in the Midday pattern, resulting in congestion forming in the early afternoon and leading to gridlock. Evidently, running a 24-hour scenario implies more challenges than running each time period separately.

Evaluating Model Results

DTA model runs produce a wide range of outputs for examination in the GUI as well as result files to be exported for post-processing and examination outside the program. Both DynusT and Dynameq save vehicle paths and aggregate simulation statistics for user-specified time intervals. Examination begins with global network and convergence measures, such as network clearance and relative gap, and proceeds to node, link and zone specific (local) results. Producing results at fine-grained intervals, such as five minutes, is desirable for visually examining outputs during network development, trouble shooting, and for other fine-grained analysis.

Checking for network clearance

Once an assignment procedure has run to completion, the first item to check is whether all vehicles have reached their destinations and left the network within a reasonable amount of time after the last demand loads. The relevant outputs are the number of vehicles waiting to enter the network and the number of vehicles left in network at the end of the simulation. Figure 13, below, shows an example of a network that does not clear at the end of the simulation period. Interval 36 corresponds to 18:00, which is when the last demand loads onto the network. Trips on the case study network should not take much more than 20 minutes, so vehicles entering the network at 18:00 could be expected to have cleared by 18:30 or 19:00. The blue line shows over 2,000 vehicles are still waiting to enter the network three hours after the last demand was loaded onto the virtual links. These vehicles are stuck on the virtual links and cannot get onto the network to complete their trips.

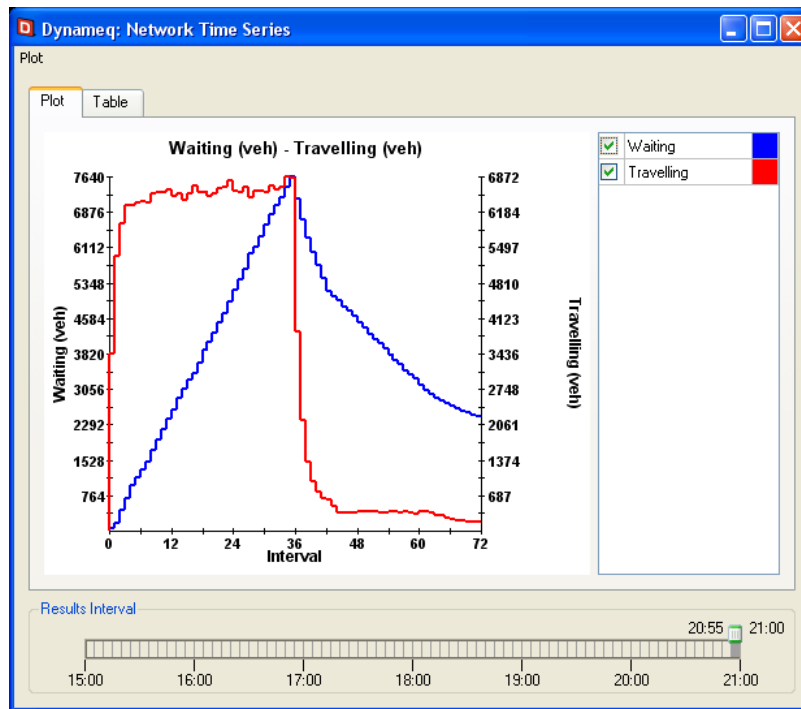


Figure 13. When the last demand is scheduled to load onto the network, this example reveals vehicles waiting to enter the network (blue) and vehicles still traveling (red) three simulation hours later.

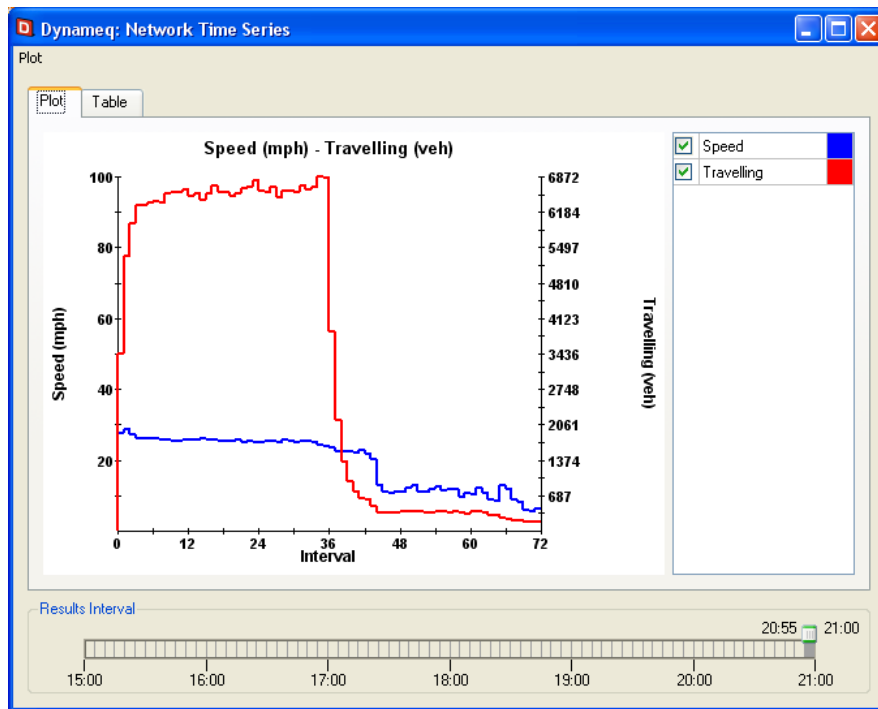


Figure 14. Plotting speeds and traveling vehicles is another way of detecting clearance problems.

For the same DTA run, Figure 14 shows average speed on the network and number of vehicles traveling at each assignment interval. Average speeds are just below 30 mph during the 15:00 to 18:00 period. After 18:00, most of the network clears except the vehicles stuck on connectors and slowly entering the network, and the average speed drops to 10-15 mph.

Relative gap

If the network clears, we want to find out whether the DTA reached an acceptable state of equilibrium. The relative gap measure is used in DTA to indicate how close an assignment is to dynamic user equilibrium (DUE). DUE is achieved when, for a given departure time interval, all used paths connecting an origin-destination pair have the same travel time, no user can improve their travel time by switching paths, and this is true for all origin-destination pairs. The relative gap statistic for the last iteration is typically of most interest. The relative gap is the percentage difference in travel times for used paths corresponding to the same departure time interval. In DTA, a relative gap of 2 percent is considered to be good, but is not easily achieved in highly congested networks. A large relative gap (e.g., greater than 5 percent) means that travel times still differ considerably between different paths for the same origin and destination pair and that the model has not achieved equilibrium. Relative gap may be expressed per departure time interval, or as an average across all departure time intervals. DynusT reports relative gaps as an average across all departure time intervals, but calculated for each interval separately. Figure 15, below, shows the average (left) and interval-specific (right) relative gaps for Dynameq. Later departures see more changes in travel times for OD pairs between iterations, as the network becomes more congested.

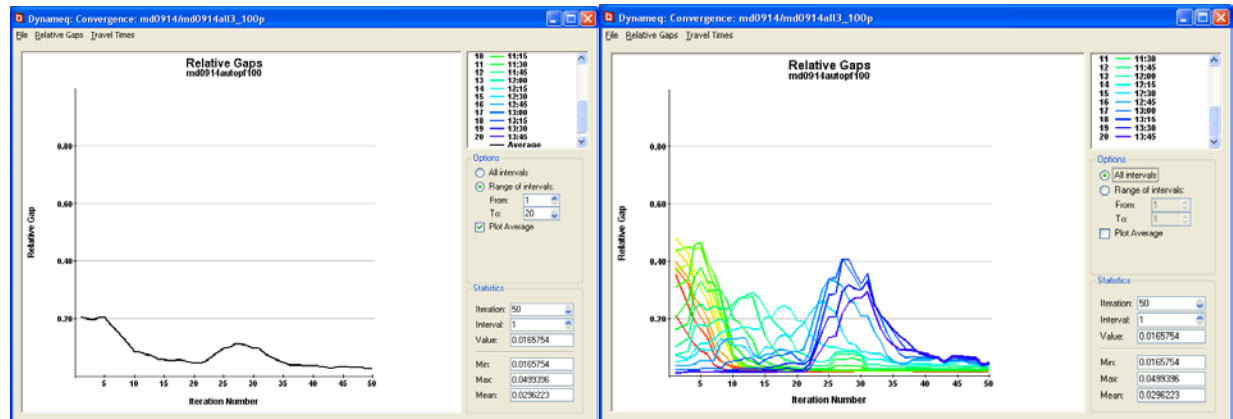


Figure 15. Relative gap average (left) and interval-specific (right) for Dynameq

In our simulation runs, both programs typically converged to similar relative gap measures for the same scenario. Where differences occurred they could usually be attributed to one program or the other producing gridlock conditions. As a note of caution, it is possible to achieve a relatively tight relative gap with a network in gridlock if congested travel times do not change between iterations; thus, relative gap is an insufficient measure of network health and should be examined after network clearance has been established.

More DTA output

DTA output can be useful during model validation and calibration, as well as in subsequent analysis runs. Basic network-wide results include average travel time and stop/wait time, average network speed per arterials and freeways. To identify network hotspots, select link analysis may be used to examine the variability of travel times for used paths between the same OD pair at different departure times and for different user or vehicle classes. Link or lane density or occupancy and queue lengths, and a range of output on the node or centroid level, are also available. For affected vehicle or multiple user class analysis, vehicles may be tagged and tracked through the network. DynusT offers the possibility of using vehicle and path information from a completed run instead of demand matrices for sensitivity testing and comparison to baseline scenarios. Subareas within the DTA network can be created for more in-depth analysis.

Validation data

The emphasis of this study is on the process of developing a DTA model, not whether one is more accurate than the other. DTA validation procedures are similar to those of static assignment macro models. To compare DTA model results to observed counts, speeds or turning movement volumes, link-level data by time interval are aggregated to match the time-interval resolution of validation data. We calculated very simple validation measures using count data for 46 arterial sites and count and speed data for 18 freeway detector locations (see Figure 16 below).

supplemented with 2009 data in cases where 2005 data were missing or had obvious errors. Wanting continuity across all count locations for a given freeway facility (US 26 or OR 217), we selected a day's worth of data for which valid counts were obtained for each detector location, which proved to be more of a challenge than we expected.

Some lane detectors reported unrealistically high speeds, and others had occasional outages due to construction or detector malfunction. The validation process also revealed network inconsistencies that could lead to misleading indicators. PORTAL data are reported as an average over the number of lanes at the count location. For validation, the count was multiplied by the number of lanes. In some instances we found discrepancies between the number of lanes in PORTAL and the DTA network. Figure 17 shows a segment of US 26, which has three real freeway lanes and one acceleration-deceleration lane. The PORTAL lane detector counts traffic only on the proper freeway lanes. The VISUM/NAVTEQ network did not discriminate between these types of lanes, and the DTA network showed four freeway lanes as a result, causing the PORTAL counts to appear unusually low by comparison before the problem was detected.



Figure 17. Difference in number of lanes. PORTAL (green line) counts three freeway lanes where the imported VISUM/NAVTEQ network (red line) has four.

A more refined validation process, with systematic quality checking and averaging, is highly recommended. Quality control of validation data is very important. Traffic volumes on the same facility vary between seasons, month and even days of the work week. For validation to count data, a range of acceptable volumes should be created for each count location, together with confidence levels that the estimated volumes would fall within the observed range.

Scenario Analysis

In this section we discuss the results of a series of testing scenarios we ran using each program, comparing the results to various validation measures. These scenarios include variations in demand loading, signal timing assumptions, and sensitivity testing of several input parameters. The main objectives of these scenarios were to gain a better understanding of how each DTA program performs under varying conditions. In doing so, we compared each program to the type of count and speed data that would normally be used in network validation. As described above, however, we have concerns and recommendations for the proper way to use the count and speed data. In addition, neither program's network and demand data has been calibrated. Therefore, we view the comparisons made in this exercise and described below not as a validation test, but rather as a demonstration of program behaviors and trends.

In order to provide a comparison among equals, we developed a set of rules to follow when setting up and running assignments:

- A single desktop computer was used for all scenario runs for both programs: a Dell Precision T7500 running Windows 7, with a 64-bit 3.33 GHz Intel Xeon processor and 24 GB of RAM.
- Every scenario was run for 50 iterations, regardless of the relative gap achieved. In most cases, both programs achieved relative gaps close to 5 percent. In a few scenarios, however, DynusT did not achieve a 5 percent relative gap by the 50th iteration.
- All scenarios were run with peak-factored demand tables.
- We ran a 1-hour warm-up period and a 1-hour cool-down period during which full demand was being loaded. We then simulated an additional two hours longer than the demand period, during which time no new demand was added.
- Count and speed comparisons do not include the warm-up or cool-down periods.
- Estimated volumes and travel speeds were obtained from the final iteration (no. 50).
- Dynameq was run using the flow-balancing, path-pruning, and dynamic path search options, as discussed above in the description of the program.

With these principles in mind, we created the following scenarios:

1. Baseline PM Peak hour 3 to 6 p.m. (2 to 7 p.m.)
2. Baseline Midday 10 a.m. to 1 p.m. (9 a.m. to 2 p.m.)
3. Baseline 24 hour
4. Saturated Midday 110%
5. Sensitivity testing on 100% Midday:
 - a. Lane closures
 - b. Default vs. realistic signal timing plans
 - c. Arbitrary vs. realistic intersection operations (DynusT only)
 - d. Arbitrary vs. realistic lane geometries (Dynameq only)

Each of these scenarios is described in greater detail below, including their setup, objectives and outcomes. One outcome that is present in all scenarios is that the total vehicles generated by each program are different from the input demand and different from each other. Differences in total vehicles generated may be attributed to two factors: (1) rounding error when converting

fractional demand values to integer values for the simulation; and (2) that DynusT assigns intrazonal trips to the network, whereas Dynameq does not.

Baseline PM Peak -- 3 to 6 p.m.

Setup

The initial objectives of this scenario were to obtain a baseline analysis for a PM Peak demand period and to work through the first three stages of network model development, as discussed previously. We began with the assumption that we would be able to load 100 percent of the demand for the time period, with full implementation of signal timing plans. We were to use this scenario as an initial comparison of the two DTA programs and as a basis for other the other scenario runs listed above. We set the three-hour period of 3 to 6 p.m. as the analysis period. With addition of warm-up and cool-down periods, the entire simulation covered the period of 2 to 7 p.m.

Outcomes

The process of making this scenario “work” is well covered in the above sections on network model development (Stages 1-3). To summarize, the network did not clear at 100 percent demand. Congestion and gridlock began at the edges of the network, despite alterations to the PM signal timing plans at some of the external stations.

We eventually settled for running the Baseline PM Peak scenario with demand reduced to 75 percent (75% autos, 75% medium trucks, 75% heavy trucks) and full signal timing, which both programs were able to clear.

A summary of model run statistics for both programs is shown in Table 1, below. The top portion of the table shows the total input demand, and the lower portion shows the summary statistics for both Dynameq and DynusT.

- Differences in total vehicles generated by the two programs are less than 3,000, about 1.5 percent of total demand. During the PM peak this relatively small percentage would suggest most trips are for commute purposes and thereby longer.
- Vehicles on the DynusT network have longer average travel times and shorter average trip distances. DynusT is assigning more trips to arterials and fewer to freeways, compared with Dynameq.
- After 50 iterations, both programs show a mean relative gap (across assignment intervals) of less than 5 percent (by vehicle class for Dynameq).
- The CPU time for Dynameq was 51 minutes, compared with 86 minutes for DynusT, about 70 percent longer.

Table 1. Summary of Model Run Statistics -- PM Peak

Demand 2 to 7 p.m.	All data from the final iteration (iter 50)	
Total demand	221273	
Auto demand	216936	
Truck demand	4337	
Average hourly demand	44255	
	Dynameq	DynusT
Total vehicles generated	219400	222160
Average travel time (minutes)	9.15	13.28
Average stop time (minutes)	3.56	4.65
Average total travel time (minutes)	12.71	17.93
Average trip distance (miles)	3.57	3.29
Total travel times excl entry queue (hours)	33460	49167
Total trip distance	782503	730737
Total stop time	13032	17208
CPU time (seconds)	3055	5186
Relative Gap		0.049
Auto	0.032	
Heavy Truck	0.046	
Medium Truck	0.048	

Comparisons of assignment volumes with observed volumes at count locations are shown in the charts and tables that follow. As shown in Figures 18 and 19, below, for Dynameq and DynusT, respectively, both programs appear to do a relatively good job of matching counts at an aggregate level. Both programs achieve R^2 values of 0.90 in the two plots. The clusters of observations in the lower left-hand corner of the plots represent arterial count locations, and the clusters of observations in the upper right-hand corner of the plots represent the higher-volume freeway locations. Upon closer inspection, it is apparent that both programs do a better job of matching freeway counts than arterial counts. Further, both programs systematically under-predict flows at all locations, as would be expected from assignment of only 75 percent of total demand.

Tables 2 and 3, below, show the ratio of estimated-to-observed demand for the Dynameq and DynusT assignment runs, respectively. The tables are broken down by arterial and freeway types, with separate sections for OR 217 and US 26. Ratios are shown for each of the three hours of the analysis period, and for the three-hour total.

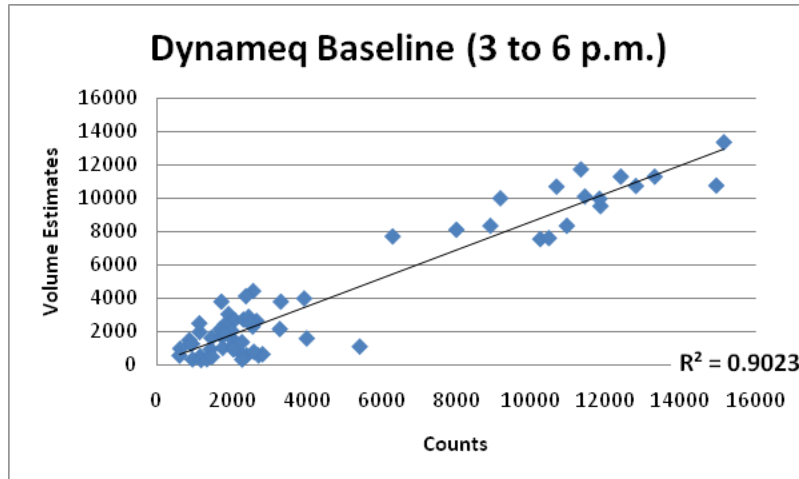


Figure 18. PM Peak observed vs. estimated traffic volumes (all locations) – Dynameq

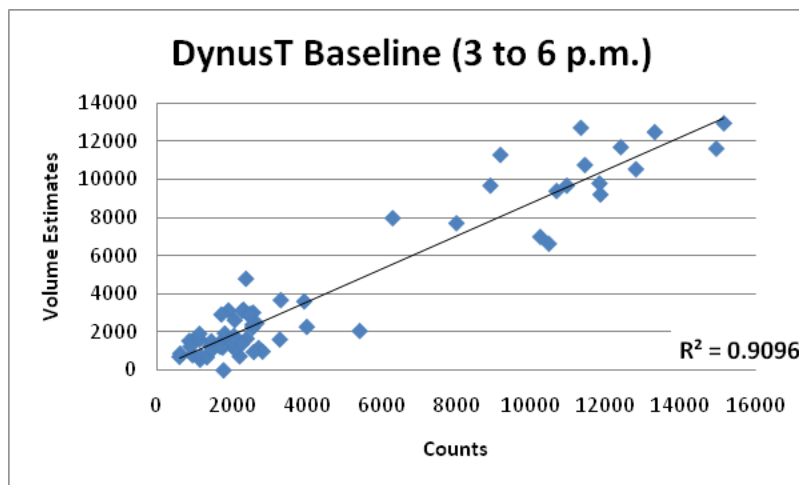


Figure 19. PM Peak observed vs. estimated traffic volumes (all locations) – DynusT

As expected for a partial demand loading, all of the ratios shown in Tables 2 and 3 are less than 1.0, but significantly higher (.87 to .89) than the 75 percent of demand that might be expected. This suggests that the true demand reflected in the counts might actually be lower than the input demand. As mentioned in the description of the third stage of network development, however, we ran a 100 percent demand PM Peak assignment without signalization and found that the sum of field counts was very close to the sum of simulated volumes across our 86 count locations, for both programs. Based on this information, we concluded that the model demand was not unrealistically high. Rather, we speculate that the additional demand in the 100 percent scenario gets dispersed over a wider range of the network than the 75 percent scenarios. Additional analysis is needed.

That the ratios are closer in the last of the three analysis hours suggests that the true demand might be more concentrated in the first hour or two than our peaking factors indicate. In

addition, it is apparent that a significantly greater share of the DynusT trips are assigned to arterials and fewer assigned to freeways, compared with the Dynameq results.

Table 2. Total Estimated/Observed – Dynameq

Facilities	Time Periods			3-hour
	3 to 4 p.m.	4 to 5 p.m.	5 to 6 p.m.	Totals
All Locations	0.844	0.857	0.912	0.872
Arterials	0.827	0.826	0.886	0.848
Freeways	0.851	0.872	0.925	0.883
OR 217	0.864	0.884	0.942	0.898
US 26	0.835	0.857	0.905	0.866

Table 3 Total Estimated/Observed – DynusT

Facilities	Time Periods			3-hour
	3 to 4 p.m.	4 to 5 p.m.	5 to 6 p.m.	Totals
All Locations	0.890	0.874	0.905	0.890
Arterials	0.860	0.841	0.904	0.869
Freeways	0.904	0.889	0.906	0.899
OR 217	0.957	0.921	0.946	0.941
US 26	0.840	0.851	0.858	0.850

Mean absolute percentage errors (MAPE) are shown in Tables 4 and 5 below for Dynameq and DynusT, respectively. Looking at totals, DynusT seems to provide a slightly better fit to the counts. At the facility level, these tables confirm that both programs better predict freeway traffic than arterial flows.

One reason for the difference between the two programs in terms of allocation of traffic to arterials versus freeways might be because we did not use an optional DynusT freeway bias factor. This option was brought to our attention by the program's developers when sharing results, but does not appear in the program's on-line documentation posted to date. The DynusT bias factor introduces the perception error describing that travelers tend to perceive freeway travel time is shorter than arterial travel time. According to DynusT developers, a factor of 20 percent tends to work well, implying that travelers perceive freeway travel time as 20 percent lower than arterials. A second source of difference may be the assignment of intra-zonal trips to the network in DynusT, which are unlikely to use freeway links, compared with no intra-zonal assignment for Dynameq.

Table 4. Mean Absolute Percentage Error (MAPE) – Dynameq				
Facilities	Time Periods			3-hour
	3 to 4 p.m.	4 to 5 p.m.	5 to 6 p.m.	Totals
All Locations	0.380	0.379	0.400	0.373
Arterials	0.467	0.469	0.503	0.465
Freeways	0.159	0.149	0.137	0.138
OR 217	0.128	0.125	0.124	0.113
US 26	0.198	0.179	0.153	0.170

Table 5. Mean Absolute Percentage Error (MAPE) – DynusT				
Facilities	Time Periods			3-hour
	3 to 4 p.m.	4 to 5 p.m.	5 to 6 p.m.	Totals
All Locations	0.339	0.316	0.329	0.320
Arterials	0.406	0.377	0.397	0.383
Freeways	0.167	0.161	0.154	0.160
OR 217	0.133	0.134	0.133	0.133
US 26	0.210	0.193	0.179	0.194

A comparison of representative freeway travel speeds may be found in Figure 20, below, which includes individual charts for each of the three analysis hours. In all three charts, the modeled speeds for the Dynameq run appear to be relatively stable compared with the observed speeds, which vary quite a bit between count locations. The modeled speeds from the DynusT run, however, are even more variable than the observed speeds, displaying relative stability at some locations and sudden drops at others. This might suggest that the DynusT network model is more sensitive to perturbations in traffic conditions; however, this sensitivity could be easily dampened by modifying link-level parameters related to speed-density relationships, a possibility discussed below in the section on calibration.

As mentioned above, moving from one hour to the next, there appears to be slightly greater changes in speeds in both observed and modeled plots for both programs, with DynusT showing the most dramatic shifts. From this we are inclined to conclude that both programs follow the general trend found in the observed data of greater congestion and slower speeds as the PM Peak period progresses.

The high average speeds shown in the PORTAL data at certain detector locations raise suspicions that the speed detectors may not be accurate everywhere. It should also be noted that in DynusT and Dynameq maximum speeds were set at 60 and 55 mph, respectively. This difference in the maximum speeds was not discovered until late in the study and is due to the fact that coded speed limits are adhered to strictly by Dynameq, but DynusT allows some vehicles to travel up to 5 miles per hour faster. Setting higher maximum speed thresholds based on observed speeds would be a good idea for calibration.

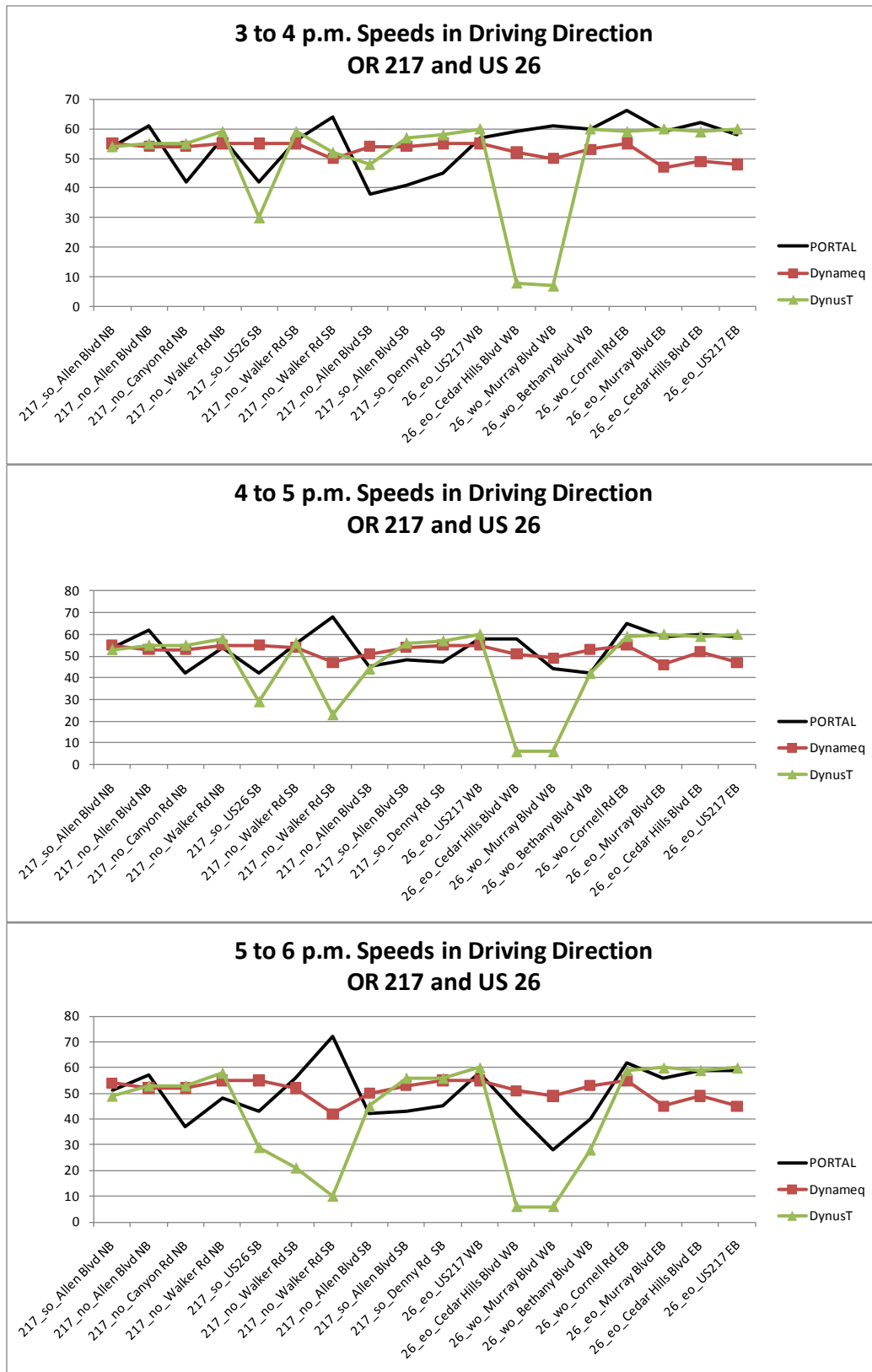


Figure 20. PM Peak observed vs. estimated freeway travel speeds by location and hour

Baseline Midday -- 10 a.m. to 1 p.m.

Setup

The initial objectives of this scenario were to provide a contrast with the PM peak scenario; however, after we were unable to accommodate full demand in the PM peak, the Midday scenario became the benchmark scenario for subsequent sensitivity analysis scenarios. The Midday scenario was run with 100 percent demand and full implementation of fixed time signal plans. We set the three-hour period of 10 a.m. to 1 p.m. as the analysis period. With addition of warm-up and cool-down periods, the entire simulation covered the period of 9 a.m. to 2 p.m.

Outcomes

Having worked out many of the kinks in the network in the PM peak scenario, the Midday scenario ran relatively smoothly. Both the models of both programs cleared and provided meaningful results. Table 6, below, shows the summary statistics.

Table 6. Summary of Model Run Statistics – Midday Baseline

Demand 9 a.m. to 2 p.m.	All data from the final iteration (iter 50)	
Total demand	207460	
Auto demand	196718	
Truck demand	10742	
Average hourly demand	41492	
	Dynameq	DynusT
Total vehicles generated	203747	208265
Average travel time (minutes)	8.38	13.38
Average stop time (minutes)	2.71	4.49
Average total travel time (minutes)	11.09	17.87
Average trip distance (miles)	3.66	3.35
Total travel times excl entry queue (hours)	28453	46437
Total trip distance	746420	698072
Total stop time	9207	15591
CPU time (seconds)	2929	4393
Relative Gap		0.068
Auto	0.03	
Heavy Truck	0.04	
Medium Truck	0.046	

- Total demand during the Midday 100 percent scenario is about 6 percent lower than the 75 percent demand of the PM peak scenario.
- In contrast to the PM Peak, the difference between the two programs in terms of vehicles generated is a slightly greater in this scenario, indicating proportionally more intra-zonal trips.
- Similar to the PM Peak, in the Midday scenario, vehicles on the DynusT network have longer average travel times and shorter average trip distances. DynusT is assigning more trips to arterials and fewer to freeways, compared with Dynameq.
- Compared with the PM Peak scenario, Dynameq travel and stop times are shorter in the Midday, but the DynusT travel and stop times are actually slightly longer, which is counter intuitive given the decrease in total demand. Both programs have longer average trip distances in the Midday than in the PM peak scenario.
- One explanation for the longer travel times for DynusT might be that the program had not converged as far by Iteration 50 as Dynameq, as indicated above by the relative gaps, which are close to 4 percent for Dynameq and nearly 7 percent for DynusT.
- CPU time are 49 minutes for the Dynameq run, compared with 73 minutes for the DynusT run, about 50 percent longer.

Comparisons of assignment volumes with observed volumes at count locations are shown in Figures 21 and 22, below. Consistent with the PM Peak scenario results, both programs appear to do a relatively good job of matching counts at an aggregate level. In this scenario, DynusT achieves an R^2 value of 0.93, compared with 0.91 for Dynameq. Given that full demand was being modeled, some improvement would be expected over the PM Peak runs.

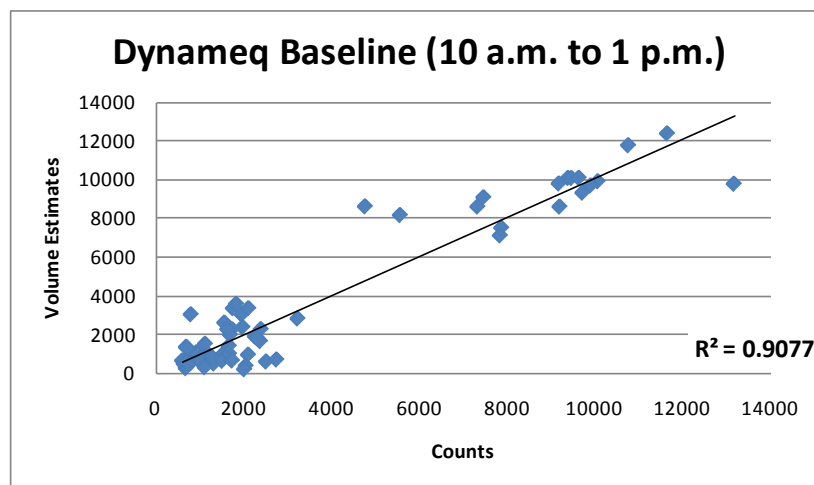


Figure 21. Midday observed vs. estimated traffic volumes (all locations) – Dynameq

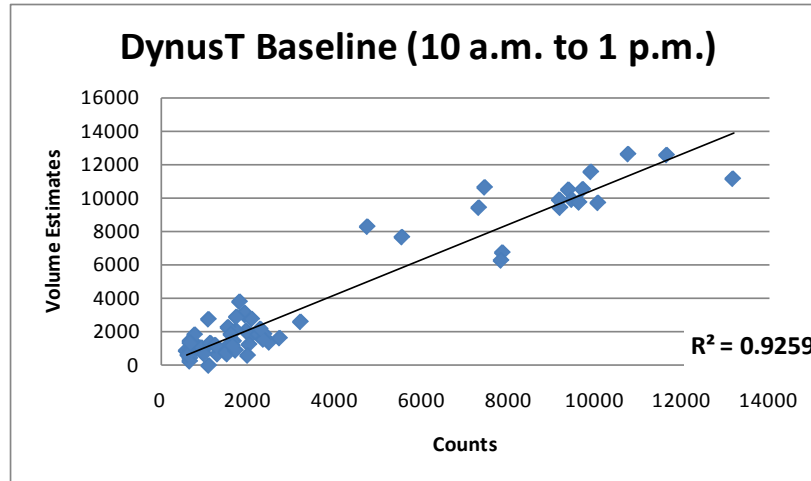


Figure 22. Midday observed vs. estimated traffic volumes (all locations) – DynusT

Tables 7 and 8 show the ratio of estimated to observed volumes by hour and by facility type. Across all locations, the ratio of 1.01 for Dynameq suggests that the modeled demand and the actual demand are a very close match. DynusT is projecting about 6 percent more demand at all count locations, which might be partially attributed to the assignment of intra-zonal trips. DynusT seems to do a better job of matching arterial count locations, whereas Dynameq seems to do a better job on freeways.

Table 7. Total Estimated/Observed – Dynameq

Facilities	Time Periods			3-hour Totals
	10 to 11 a.m.	11 to 12 p.m.	12 to 1 p.m.	
All Locations	0.942	1.007	1.065	1.008
Arterials	0.885	0.912	0.947	0.918
Freeways	0.965	1.049	1.119	1.047
OR 217	0.917	0.984	1.060	0.990
US 26	1.027	1.136	1.198	1.123

Table 8. Total Estimated/Observed – DynusT

Facilities	Time Periods			3-hour Totals
	10 to 11 a.m.	11 to 12 p.m.	12 to 1 p.m.	
All Locations	1.044	1.063	1.063	1.057
Arterials	0.977	0.980	1.019	0.994
Freeways	1.070	1.099	1.084	1.085
OR 217	1.067	1.077	1.079	1.074
US 26	1.075	1.129	1.091	1.099

The MAPE statistics shown below in Tables 9 and 10 reflect these same differences between the two programs. Compared to the PM Peak scenario, both programs actually have higher count error in the Midday. This seems counter intuitive given the better match in total demand, but this might be because in the Midday we are both over and under-estimating volumes whereas in the PM peak there was predominately under estimation. Consistent with the PM peak, both programs assign more vehicles to the last of the three analysis hours, compared with the counts. Again, this could suggest that peaking factors need adjustment.

Table 9. Mean Absolute Percentage Error (MAPE) – Dynameq				
Facilities	Time Periods			3-hour
	10 to 11 a.m.	11 to 12 p.m.	12 to 1 p.m.	Totals
All Locations	0.416	0.409	0.403	0.397
Arterials	0.525	0.508	0.484	0.492
Freeways	0.140	0.155	0.194	0.153
OR 217	0.114	0.106	0.131	0.106
US 26	0.173	0.216	0.272	0.212

Table 10. Mean Absolute Percentage Error (MAPE) – DynusT				
Facilities	Time Periods			3-hour
	10 to 11 a.m.	11 to 12 p.m.	12 to 1 p.m.	Totals
All Locations	0.401	0.359	0.363	0.360
Arterials	0.489	0.423	0.431	0.428
Freeways	0.175	0.196	0.188	0.186
OR 217	0.149	0.143	0.168	0.154
US 26	0.207	0.262	0.214	0.226

Estimated and observed freeway travel speeds for the Midday scenario are shown below in Figure 23. The results are similar to the PM Peak scenario. Although the average travel speeds from the count detector locations vary quite a bit by location, they remain stable across all three hours. The Dynameq freeway travel speeds are also very stable across all three hours, but do not vary much across locations either. In contrast, the DynusT travel speeds exhibit large fluctuations across both detector locations and hours of analysis. These results reinforce the results from the PM peak scenario, that DynusT is more sensitive to perturbations in traffic flow, which seem to recur at certain locations. Further investigation of these locations would be needed to pinpoint the source of the problem and whether link-based traffic flows need to be adjusted.

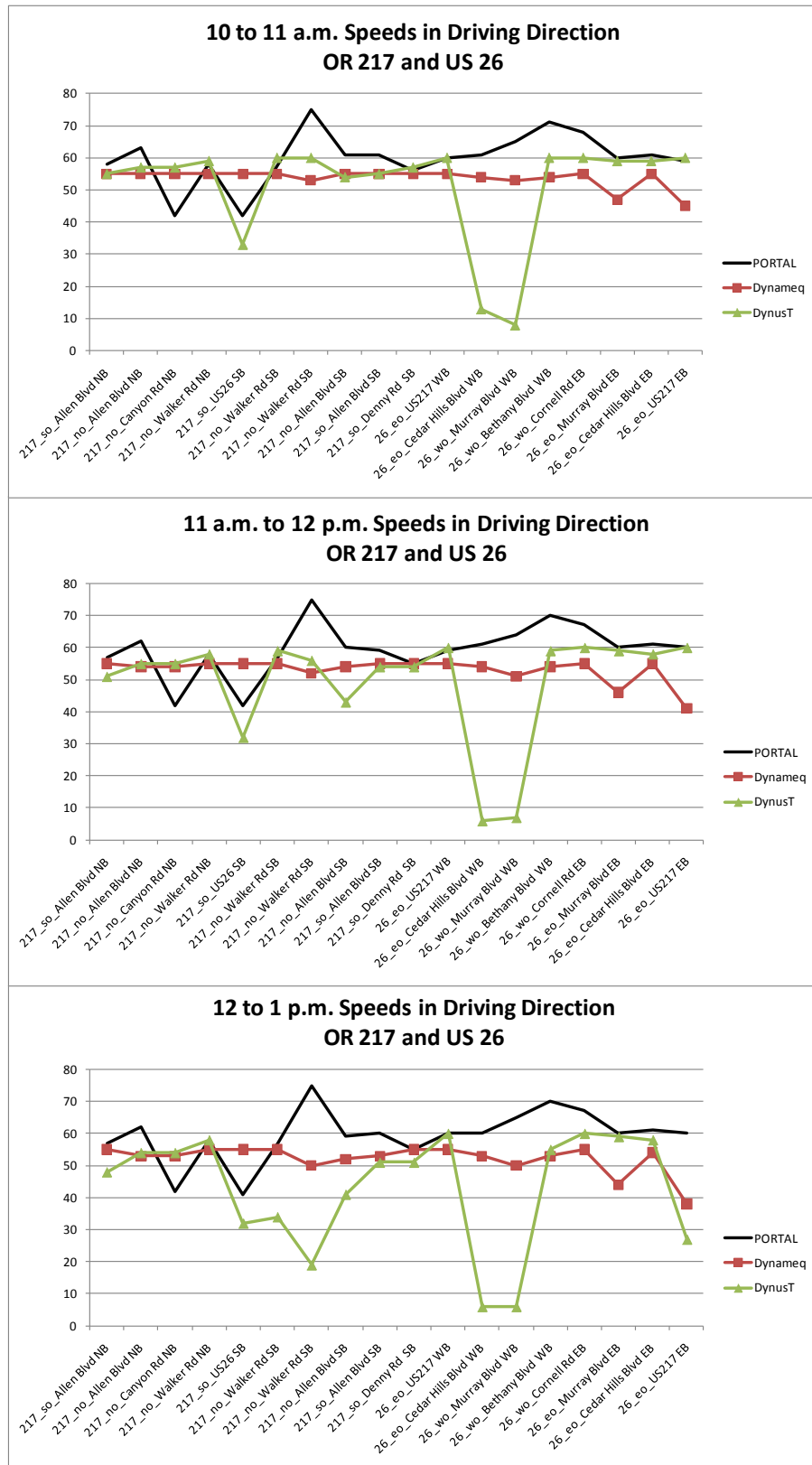


Figure 23. Midday observed vs. estimated freeway travel speeds by location and hour

24-Hour – 1 a.m. to 11 p.m.

Setup

The initial objectives of this scenario were to test whether a 24-hour scenario could be run and how the results would differ from the three-hour runs during the same time periods as the Midday and PM Peak scenarios. We created demand tables from the eight original Metro assignment periods, combining them and applying the diurnal peaking factors for every 15-minute interval of a 24-hour day.

We recognized that post-assignment analysis would not consider the first and the last hour of demand, since those were needed for warm-up and cool-down, respectively. Neither program's GUI would allow an analysis of extending beyond 24 hours.

Since the PM Peak scenario with signalization was not able to clear with full demand in either program, we considered two feasible options: (1) run 75 percent demand for all time periods with full signalization, knowing that the PM Peak had cleared with this level of demand; or (2) run 100 percent demand for all time periods, but without signalization.

In addition, the DynusT GUI was not able to use timing plans for multiple time periods, limiting signal timing plans to a single set for the entire day. This meant that we would only try a 75 percent signalized scenario in Dynameq.

Outcomes

We ran the signalized scenario with 75 percent all-day demand in Dynameq. Despite having successfully run AM Peak signalized and Midday signalized scenarios with higher demand, and having successfully run a PM peak scenario with 75 percent demand, this 24-hour variation formed total gridlock at about 3 p.m. in the simulation. Apparently, travel patterns and congestion formed in the 24-hour scenario are different from those created by a one-hour warm-up period and create additional challenges. We speculated that the switch from a set of Midday timing plans to PM Peak timing plans, at a time of increasing demand, may have shocked the system, but further investigation is warranted.

We then tested the 100 percent demand with no signalization in both programs. This scenario completely cleared in DynusT, but not in Dynameq. Wanting to compare equals, we decided to move forward with a “safer” 75 percent unsignalized scenario, which cleared in both programs. Table 11, below shows the summary statistics for the 24-hour run with 75 percent demand and no signalization.

- The average hourly demand is nearly half that of either the PM Peak (75%) or the Midday (100%) scenarios described above.
- An important difference from the previous scenario results is that there remain vehicles in the network at the end of the simulation period. This is because we were not able to run the simulation for an extra hour or two without demand after the 24th hour in order to let all of the vehicles clear. Nevertheless, both networks appeared to be well on their way to clearing.

- Similar to the PM Peak and the Midday scenario, vehicles on the DynusT network have longer average travel and stop times and shorter average trip distances. DynusT is assigning more trips to arterials and fewer to freeways, compared with Dynameq.
- In both programs travel and stop times are significantly shorter than in either the PM Peak or Midday scenario runs, while average trip distances are about the same. This would be expected since the 24-hour runs include night and early morning trips during uncongested conditions.
- Similar to the PM Peak and Midday periods, one explanation for the longer travel times for DynusT might be that the program had not converged as far at Iteration 50 as Dynameq, as indicated above by the relative gaps, which are 5 percent for Dynameq and nearly 10 percent for DynusT.
- CPU time are 158 minutes for the Dynameq run, compared with 215 minutes for the DynusT run, about 36 percent longer.

Table 11. Summary of Model Statistics -- 24-Hour Run

Demand 24 hours	All data from the final iteration (iter 50)	
Total demand	579,756	
Auto demand	560,868	
Truck demand	18,888	
Average hourly demand	24,157	
	Dynameq	DynusT
Vehicles in network at end	642	334
Vehicles waiting to enter network at end	181	
Total vehicles generated	570208	579963
Average travel time (minutes)	5.96	7.70
Average stop time (minutes)	0.51	0.67
Average total travel time (minutes)	6.47	8.37
Average trip distance (miles)	3.59	3.33
Total travel times excl entry queue (hours)	56646	74403
Total trip distance	2045703	1930294
Total stop time	4872	6516
CPU time (seconds)	9503	12900
Relative Gap		0.097
Auto	0.005	
Heavy Truck	0.005	
Medium Truck	0.005	

Since this 24-hour run was run with 75 percent of total demand, we felt it would be most instructive to examine how estimated volumes tracked observed volumes. Figures 24 and 25, below, show a general trend in which both DynusT and Dynameq simulations seem to capture the general diurnal pattern quite well on both arterials and freeways, respectively. Despite representing 75 percent of demand, the DynusT estimated volumes were actually at 90 percent of the observed volumes across all locations, and the Dynameq were at 77 percent. Both programs were closer in the AM and PM peak periods. The difference between the two programs is noticeable, particularly in the Midday, but the reasons for this difference are unclear.

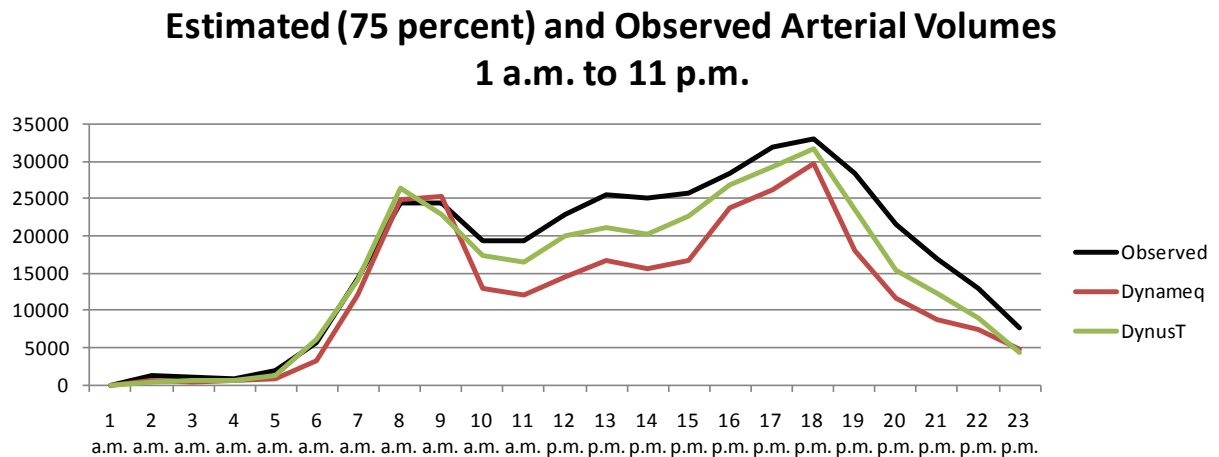


Figure 24. A “24-hour” unsignalized run with 75% demand estimated volumes at 46 arterial locations

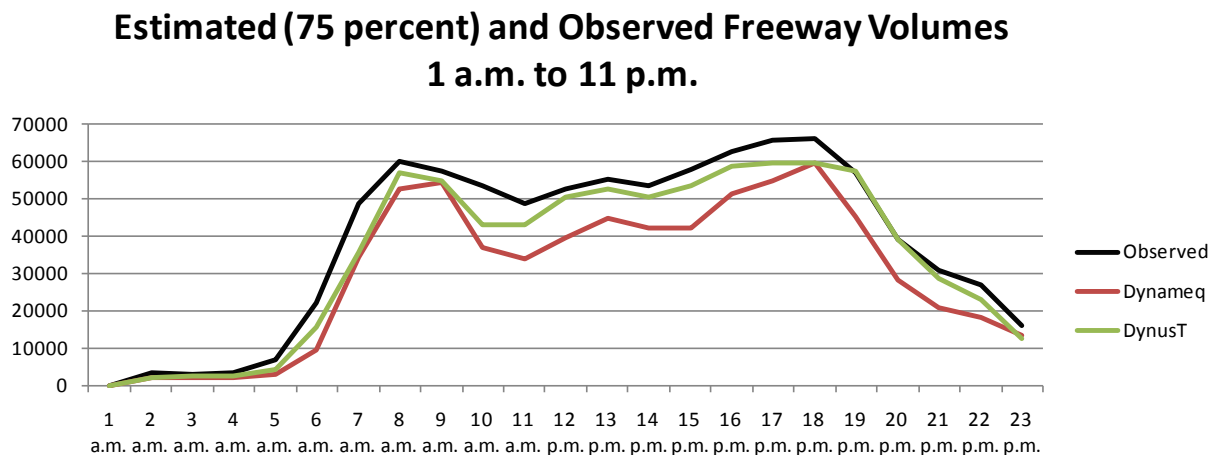


Figure 25. A “24-hour” unsignalized run with 75% demand estimated volumes at 18 freeway locations

Saturated Midday (110%)

Setup

The 2035 Regional Transportation Plan assumes an increase in Portland tri-county/metropolitan area population from 1.4 million to 2 million, a 40 percent increase. As a “stress test” we wanted to see whether we could increase the Midday demand to this level while still clearing the network at the end of a simulation.

As in the baseline, we implemented signal timing plans and set the three-hour period of 10 a.m. to 1 p.m. as the analysis period. With addition of warm-up and cool-down periods, the entire simulation covered the period of 9 a.m. to 2 p.m.

We first ran the Midday scenario with 140 percent demand, but the network did not clear. Subsequently, we gradually reduced the demand in ten percent decrements until the network cleared, at 110 percent of baseline demand.

Outcomes

Results from the 110 percent Midday run are summarized in Table 12, below.

- Average travel and stop times are greater in both programs, compared with the Baseline Midday scenario, as would be expected with greater demand. Distances, however, are nearly the same.
- Both programs achieved similar relative gaps, in the 5 percent range for Dynameq and 6 percent for DynusT.
- The CPU time for Dynameq was 51 minutes. DynusT’s CPU time was 86 minutes, 67 percent longer. These times were both slightly longer than those obtained in the baseline run.

Figure 26, below, shows the travel speeds obtained for each of the 18 freeway detector locations for all three hours of the Midday period. The observed speeds, although based on 100 percent demand, are included for comparison. The estimated speeds for both programs for both the Baseline Midday and the 110 percent Midday scenario are also shown. The results of the 110 percent runs are similar to the 100 percent runs for both programs. Specifically, Dynameq continues to show relatively stable freeway speeds, with only a slight dip in travel speeds at a few count locations. In contrast, DynusT shows even more dramatic drops in travel speeds than in the baseline, again suggesting a greater sensitivity.

Table 12. Summary of Model Statistics -- Midday 110% Demand

Demand 9 a.m. to 2 p.m. 110%	All data from the final iteration (iter 50)	
Total demand	228206	
Auto demand	216390	
Truck demand	11816	
Average hourly demand	45641	
	Dynameq	DynusT
Vehicles in network at end	0	1202
Total vehicles generated	224631	228226
Average travel time (minutes)	9.17	17.49
Average stop time (minutes)	3.47	5.68
Average total travel time (minutes)	12.64	23.17
Average trip distance (miles)	3.67	3.34
Total travel times excl entry queue (hours)	34343	66538
Total trip distance	825491	763041
Total stop time	12980	21593
CPU time (seconds)	3080	5153
Relative Gap		0.06
Auto	0.051	
Heavy Truck	0.045	
Medium Truck	0.049	

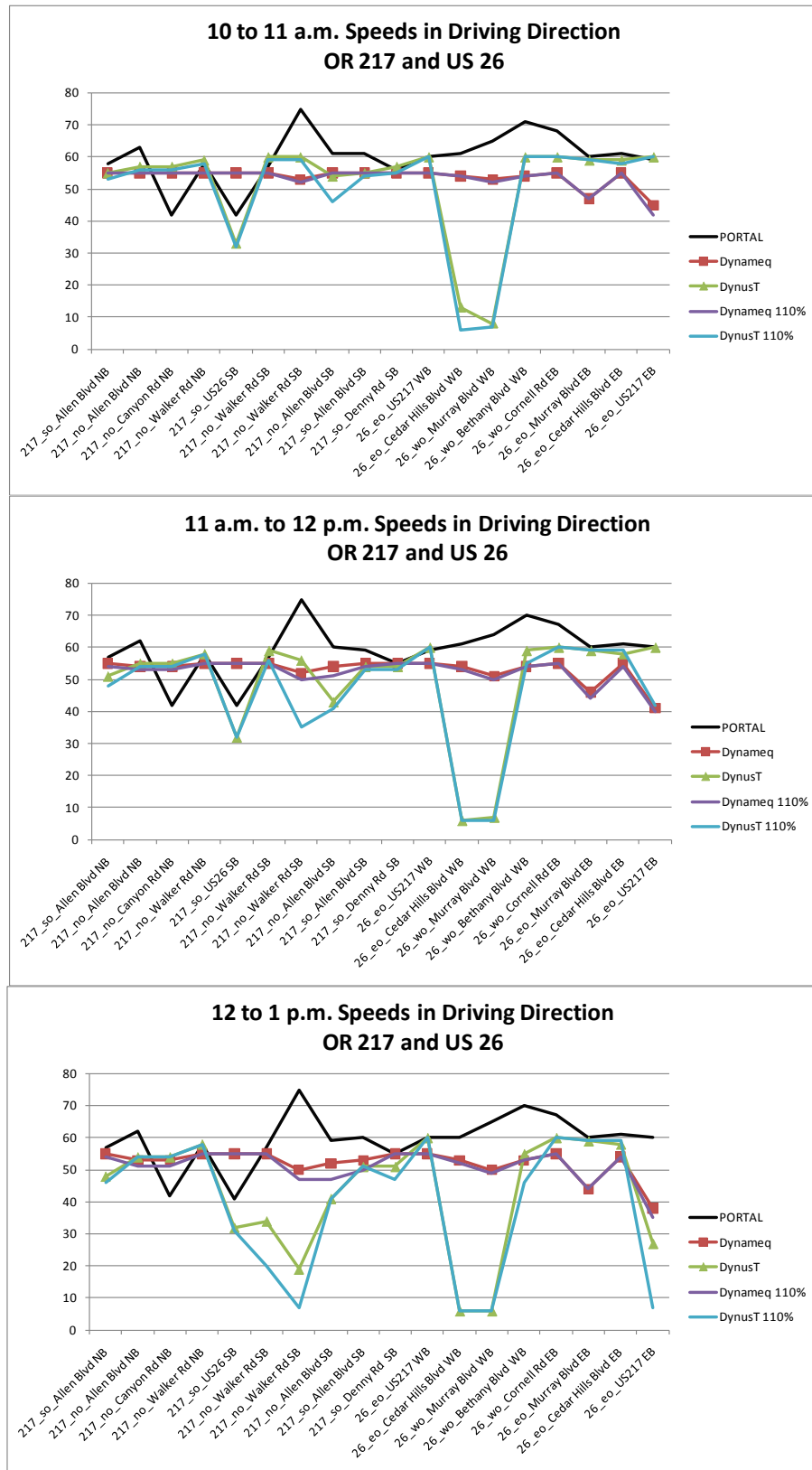


Figure 26. Midday observed vs. estimated (110% demand) freeway travel speeds by location and hour

Sensitivity tests using Baseline Midday -- 10 a.m. to 1 p.m.

We chose to perform a limited amount of sensitivity tests on the Midday scenario, as it was the only signalized scenario handling 100 percent of the demand. While numerous sensitivity tests could be performed, we chose a few tests that might help to extend our knowledge on some of the issues discussed above. In particular, we were interested in testing the impacts on network performance of variations in link capacity and signal timing assumptions.

Default vs. realistic signal timing in Dynameq and DynusT

For comparison with realistic signal timing plans, we created a Midday scenario using Metro's default timings (25 seconds green, 4 seconds amber, and 1 second all-red). Summary statistics are shown below in Table 13.

Table 13. Summary of Model Run Statistics

Demand 9 a.m. to 2 p.m.	All data from the final iteration (iter 50)			
Total demand	207460			
Auto demand	196718			
Truck demand	10742			
Average hourly demand	41492			
	Dynameq		DynusT	
	Realistic	Default	Realistic	Default
Vehicles waiting to enter network		552		
Vehicles in network at end		132	398	92
Total vehicles generated	203747	203839	208265	207499
Average travel time (minutes)	8.38	10.02	13.38	14.71
Average stop time (minutes)	2.71	4.12	4.49	5.69
Average total travel time (minutes)	11.09	14.13	17.87	20.40
Average trip distance (miles)	3.66	3.72	3.35	3.41
Total travel times excl entry queue (hours)	28453	34030	46437	50883
Total trip distance	746420	759200	698072	706951
Total stop time	9207	13986	15591	19663
CPU time (seconds)	2929	3022	4393	4845
Relative Gap			0.068	0.05
Auto	0.03	0.06		
Heavy Truck	0.04	0.09		
Medium Truck	0.046	0.11		

- Using default signal timing plans, average travel and stop times and trip lengths increased in both programs, compared with the realistic signal timing scenario.
- The default Dynameq scenario suffered additional problems, as over 500 vehicles were still waiting to enter the network at the end of the simulation period (two hours after the last demand interval).
- The realistic scenario achieved a smaller relative gap in the Dynameq run, whereas the default scenario had a smaller relative gap in the DynusT run.

Table 14 below shows the ratio of estimated-to-observed volumes on arterials for both programs. Table 15 shows the mean absolute percentage error (MAPE) on arterials. As expected, the DynusT run with default timings did not match arterial counts as well as the scenario with real signal timing plans. While the realistic signal timing scenario matched counts at a total of 99 percent for the three hour analysis period, the default timing scenario ratio of estimated-to-observed volumes was 95 percent. The difference is especially noticeable between 11 a.m. and 12 p.m. As expected, the difference on freeways (not shown) was much smaller (default estimated/observed was 1.085 compared with 1.072 for the realistic timing scenario). The MAPE actually indicated a slightly better fit to the counts in the default scenario compared with the realistic scenario.

According to ratio of estimated-to-observed volumes, the Dynameq run with default timing plans actually resulted in a better fit to the arterial count data, compared with the realistic timing plan scenario. In addition, the scenario with realistic timing plans resulted in a better fit to the freeway data (default was 1.001 compared with 1.047 for the realistic timing scenario). The MAPE statistics tell a different story. As shown in Table 15, arterial counts have a lower error in the realistic scenario than in the default timing scenario.

Table 14. Est/Obs -- Midday scenario with realistic and default timing plans

Arterial Estimated / Observed				
	Dynameq		DynusT	
	Realistic	Default	Realistic	Default
10 to 11 a.m.	0.885	0.942	0.977	0.948
11 to 12 p.m.	0.912	0.925	0.980	0.917
12 to 1 p.m.	0.947	0.966	1.019	0.981
3-hour Totals	0.918	0.945	0.994	0.950

Table 15. MAPE -- Midday scenario with realistic and default timing plans

Arterial Mean Absolute Percentage Error				
	Dynameq		DynusT	
	Realistic	Default	Realistic	Default
10 to 11 a.m.	0.525	0.584	0.489	0.483
11 to 12 p.m.	0.508	0.517	0.423	0.421
12 to 1 p.m.	0.484	0.534	0.431	0.400
3-hour Totals	0.492	0.521	0.428	0.416

To synthesize, this test provided somewhat inconclusive results. On one hand, the network-level data indicate that, in both programs, travelers are finding shorter paths and encountering less delay in the scenario that utilizes the realistic signal timing plans. On the other hand, it is not clear which scenario matches observed volumes better. It may be the case that real-world travelers, on average, possess a poorer perception of travel times than these two programs are portraying, in which case additional calibration measures should be considered.

Realistic, default and fully-actuated intersection operations (DynusT only)

As a follow up to the first sensitivity test described above, we decided to explore the possibility of the option in DynusT to code signals as fully actuated. If the fixed signal timing creates inefficiencies in the network, with unused green time in places and insufficient green time in others, actuated signals may reduce these inefficiencies. This may also be the type of signal timing that one would need to be use in a future-year scenario in which timing plans are unknown, and demands and capacities are substantially different from today.

To test this, we compared the Midday scenario with fixed timing to a Midday scenario where all signals were coded as actuated with maximum and minimum green times for all approaches. We used DynusT's default actuated settings, which set maximum green time to 40 seconds and minimum green time to zero seconds. A zero-second minimum green time implies a phase will be skipped if there is no demand in the approaching lanes.

The actuated scenario did achieve a slightly better relative gap by Iteration 50; however, this is by no means an indicator of a better assignment. Contrary to expectations, the scenario with actuated timing performed worse than the default timing, and much worse than the realistic timing plans in terms of matching counts. As shown in Table 16, below, the average travel and stop times and trip lengths were longer than in both other scenarios for the Midday. The actuated scenario also matched arterial counts less closely than the default scenario, as shown in Table 17.

While the actuated timing plans performed relatively poorly in this very limited test, before drawing conclusions, it would be instructive to test different maximum and minimum green times. It is also possible to apply signal actuation on an intersection-by-intersection basis, using different maximum and minimum green times, while leaving other intersections with fixed-time plans. Both of these options should be explored further.

Table 16. Summary of Model Run Statistics

Demand 9 a.m. to 2 p.m.	All data from the final iteration (iter 50)		
Total demand	207460		
Auto demand	196718		
Truck demand	10742		
Average hourly demand	41492		
DynusT			
	Realistic	Default	Actuated
Vehicles waiting to enter network			
Vehicles in network at end	398	92	90
Total vehicles generated	208265	207499	208196
Average travel time (minutes)	13.38	14.71	15.67
Average stop time (minutes)	4.49	5.69	6.33
Average total travel time (minutes)	17.87	20.40	22.00
Average trip distance (miles)	3.35	3.41	3.42
Total travel times excl entry queue (hours)			
	46437	50883	54380
Total trip distance	698072	706951	711054
Total stop time	15591	19663	21954
CPU time (seconds)	4393	4845	4613
Relative Gap	0.068	0.05	0.047

Table 17. DynusT Midday scenario with realistic, default and actuated timing plans

Arterial Estimated / Observed			
	Realistic	Default	Actuated
10 to 11 a.m.	0.977	0.948	0.936
11 to 12 p.m.	0.980	0.917	0.920
12 to 1 p.m.	1.019	0.981	0.969
3-hour Totals	0.994	0.950	0.943

Lane closures

A typical DTA application is estimating the local effects of a network change, such as a lane closure. We created a scenario identical to the Midday scenario, but closed the innermost lane to traffic on a three-lane stretch of US 26. In Dynameq, this link was split, so that the on- and off-ramps did not interfere with the lane closure. In DynusT, one lane was simply removed from the network, as network events cannot be lane-specific.

We then compared link data for the freeway link immediately upstream from the lane closure. In Figure 27 average inflow (vehicles per hour) and link travel times are plotted in Dynameq for the upstream link. The left plot (regular scenario with downstream link at full capacity) shows traffic travels at free-flow speeds with steady link travel times (40 seconds) throughout the study period (note that speeds are capped at 55 mph). The right plot (lane closure scenario with downstream link at reduced capacity) clearly reveals the effects of the downstream lane closure. The red spike represents the time period between 12 and 1 p.m., where speeds decrease and the link travel time triples. Inflow remains roughly the same throughout both study periods, implying that the lane closure does not produce enough upstream congestion to make drivers choose other routes. More severe delays are needed to make people choose arterial routes instead of the freeway, and there are no good alternative routes paralleling US 26 for any substantial distance.

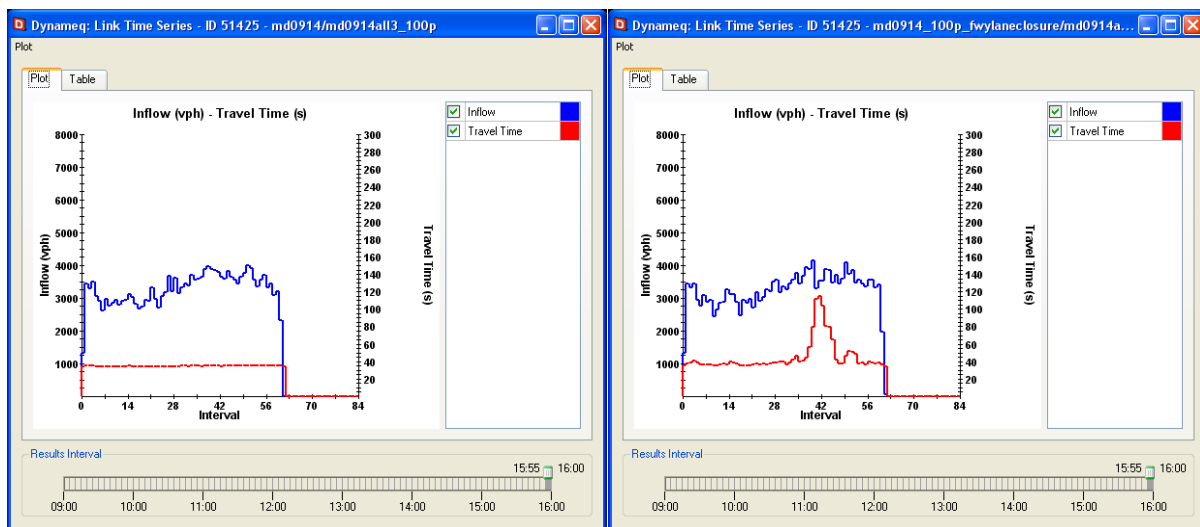


Figure 27. Dynameq portrayal of impact on upstream link of a lane closure before (left) and after (right): link volumes are shown in blue and travel times in red.

The lane closure effects are more dramatic in DynusT, as shown below in Figure 28. The top graph shows speeds on the link in the regular scenario. The downstream lane closure causes crawling speeds on the upstream link throughout most of the five hour demand period. Without more detailed sensitivity analysis, it is not clear why DynusT reacts so much more dramatically to the lane closure. One possibility is that the default values used to represent the parameters of the traffic flow model in DynusT should be adjusted for this and perhaps other links. Another possibility is that the representation of individual lanes in Dynameq allows simulated travelers to adjust to the closure more readily.

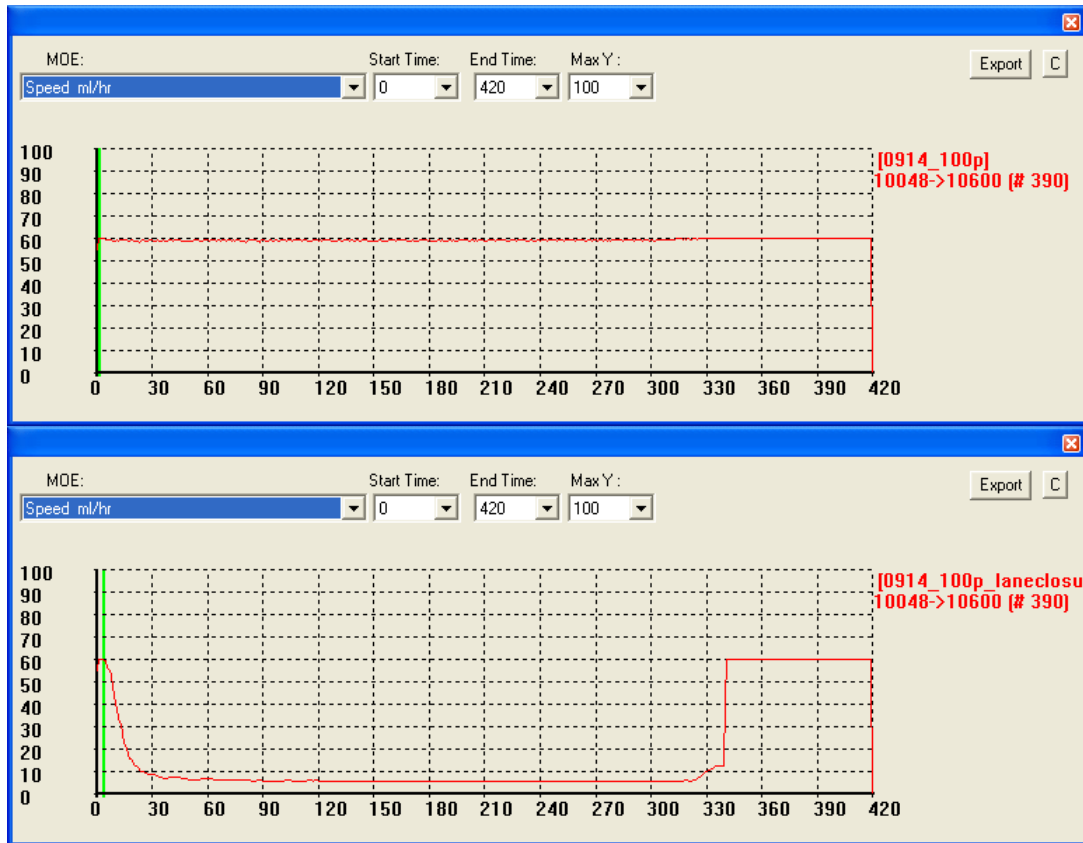


Figure 28. DynusT portrayal of impact on upstream link of a lane closure before (top) and after (bottom): the red line represents speed.

Arbitrary vs. realistic lane geometry (Dynameq only)

In the base network, turn bays typically have default lengths (200 feet in DynusT) or an arbitrary length resulting from link splitting in Dynameq. As a result, turn pockets are generally too long in the base network. This has the potential to affect the simulation, since turn bays will have more capacity than in reality, allowing vehicles to enter turn bays further upstream and potentially reducing queues on links.

Using aerial photos, we coded realistic turn bay lengths at the four adjacent intersections shown in Figure 29, below: Murray and Walker, Murray and Bowerman, Murray and Jenkins, and Jenkins and Jay. This test was specific to Dynameq because DynusT turn-bay lengths are a global link attribute and cannot be changed individually. The resulting corrections are shown below in Figure 30.

The results of the DTA run after these changes were made showed no noticeable system-wide impact. At each intersection, there were the expected changes to the turn bay links themselves—higher densities and slower speeds—but overall intersection performance was not affected in any appreciable way. Apparently, the generous capacities of the original, arbitrary coding were not missed. We assume that if we were to make similar to corrections to all of the turn bays in the region, some system-wide effect may emerge.

Next Steps and Calibration

Before calibration, the network used in this study would need to be refined and improved further. In particular, the connector layout should be improved as discussed earlier. Several zones may need to be divided to spread traffic more evenly on the network. Ramp meters should be added, and the maximum speeds adjusted upward on freeways. In the Dynameq network, turn bay lengths should be adjusted to real lengths. The data used for validation should also be improved as discussed earlier.

After these steps, remaining network coding errors, zone system problems and demand problems are typically detected in the calibration phase through select link analysis for OD pair links with large differences between simulated flows and observed counts.

Demand matrix calibration is an important part of calibration, but may also be a necessary step towards accommodation of full demand for certain scenarios, such as the PM peak scenario in this study. The reason for performing matrix calibration is a belief that the initial time-dependent demand tables are not reflective of actual demand in localized areas. Demand matrix calibration uses local counts to shift the demand in time and create an initial demand matrix that more accurately reflects actual changes in demand over the analysis period. Assigned volumes are compared to traffic counts and the timing of demand loadings is iteratively shifted until the flows more closely match counts.

Further, turning movement counts at intersections and arterial speed data should be obtained and used for calibration. Driver behavior and link performance functions such as driver response time and effective vehicle length, service flow rate, jam density and gap acceptance may need to be adjusted for different vehicle classes, user categories and facility types.

As described in the section on program descriptions, both DynusT and Dynameq developers have developed calibration tools that may help in these efforts.

Evaluation of User Experience

The objective of this study was to compare the two DTA programs. It was necessary to keep the demand and network inputs as similar as possible to enable comparisons. Developing parallel models in two DTA programs was helpful for problem solving, as the two programs picked up different input and coding problems. Nevertheless, maintaining inputs for the two programs did not leave enough time to calibrate either model. Had we spent a significant amount of time attempting to calibrate either model, the more dissimilar the two would have become, and we had concerns that calibration could lead to spending more time with one program than the other. The similarity requirement also meant that we were unable to take full advantage of the unique characteristics and abilities of each of the two programs. For example, we did not use the previously described VISUM-DynusT conversion tool and VISUM plug-in.

The reader should keep in mind that neither DynusT nor Dynameq is shown at its very best in this report. The DTA developers are better equipped to showcase the strengths and many useful features of each program. Our experience is more representative of what an agency might expect if they were to implement a DTA model on their own, with a minimum amount of assistance from the developers. With these caveats in mind, this section evaluates the user experience of this study; discussing data file handling, user interface and error reporting, documentation and technical support.

Data file handling (input and output)

Both programs take text files as input, which can be exported from the macro model and reformatted as necessary. Both programs manage scenarios through hierarchies of file folders. DynusT stores input/output file folders in text file format, which makes them easily accessible to external scripting languages and database import. Dynameq stores most files in binary formats that can only be read in Dynameq or EMME. All network, demand and results files can be exported as text or dbf files through the GUI, if desired.

Neither program yet provides the user with the option of using a scripting language or batch processing. For single assignments, it is easy to work through the GUI to set up new scenarios, execute the simulations, and collect and export the results. If the programs were to be used for a large number of runs or together with a demand model, it would be useful to have an established application program interface (API) and flexible scripting language.

User interface and error reporting

While both programs appear to have similar analytical capabilities, the most obvious difference the user of these two programs is likely to experience concerns their graphical user interfaces. INRO develops and maintains the Dynameq GUI internally. In contrast, DynusT's GUI (NEXTA) is developed and maintained externally by a professor at the University of Utah. The Dynameq GUI, while not flawless, is easy to work with. This is especially true in Dynameq 2.0, which was released during the latter stage of this study. The user can save preferred display and viewing settings for later use, which is very helpful and saves time.

The DynusT GUI, NEXTA is rather cumbersome and makes network development and model results visualization more difficult than necessary. NEXTA is buggy and very sensitive to file input formats. Further, some tools and functions are listed in the GUI even though they are not currently enabled, causing confusion.

An important note about DynusT's GUI user experience is the previously mentioned VISUM-DynusT conversion tool. Because it was not available until a later stage of this project, and in order to keep the two programs comparable, we did not use the VISUM-DynusT conversion tool in this project. The conversion tool and its VISUM plug-in appears to overcome many of the negative aspects of NEXTA by allowing network editing and results to be displayed in VISUM.

Good error reporting is very important during file import and network building. Dynameq error reporting is helpful and exhaustive; all errors of a certain type are reported at once, facilitating systematic correction and recoding. NEXTA, on the other hand, often produces unhelpful or too-general error messages, leaving the user searching for the root of the problem like a needle in a haystack. Moreover, DynusT sometimes reports errors one at a time, which hinders systematic problem solving. NEXTA performs some basic network coding error checking, which is somewhat helpful. Other errors are often not caught until the simulation is started, however, at which point the program crashes with an error message. In contrast, the Dynameq simulation will not run if certain critical errors occur.

In both programs changes made in the GUI are automatically updated in the scenario network files and can be exported from both programs. This is very helpful for network features that are edited in the GUI but need to be exported for later use as inputs, such as creating turn bays and realigning links.

Documentation

Documentation for both programs has improved throughout this project. Dynameq's documentation and user manual and demonstration project file answer many questions. All input and output files are described in great detail. DynusT's user manual is an online wiki, which has been updated, expanded and improved several times during the course of this project, including better description of all input and output files.

Technical support

Both DTA developers provided excellent technical support during this project. Dynameq communication was through email to INRO's support center. Each email had a support ticket number, to keep track of the communication. We communicated mainly with one staff member, who got to know our project well. Support was quick (typically within 24 hours) and thorough. Large scenario files were uploaded to INRO via FTP.

During most of the DynusT trial, support was provided via email and occasionally by telephone with the developer and the main research engineer. Towards the end of the project, DynusT was released as open-source software, and a support ticket system was established to replace emailing. While the email communication typically worked well with quick responses, the support ticket system was better, allowing both sides to keep track of the communication. Large

files could be uploaded via FTP. As mentioned, the DynusT GUI was developed and maintained by a different developer at a different university. This was less than ideal, because problems related to the GUI could not be solved directly by the DynusT developers, and could at best be added to a list of needed fixes that was sent to the GUI developer occasionally.

Considerations for Choosing a DTA Program

Transportation planners face a growing list of problems and tasks that require new tools like DTA. DTA programs have developed rapidly in recent years and continue to do so. With a growing number of users, more valuable suggestions for program improvements are fed back to DTA developers, who respond with easier-to-use programs. During the course of this study, we have witnessed substantial improvements to user-end valuable components such as GUIs, input and output conversion, compatibility with other programs and documentation. DTA programs are maturing and becoming increasingly user friendly and viable for network planning and operations analysis.

This report has aimed to demonstrate key aspects of DTA and, through experiments with two DTA programs, ways in which available programs can differ. The following is a summary of considerations, based on lessons learned from this project, for choosing a DTA program and planning for developing a DTA network and model.

Creating a DTA model is time consuming. Some form of automated conversion is necessary. This is especially true for the parts of a STA model network that need to be modified for DTA, such as connectors and division of zones. No matter how detailed the STA network is, and even with an automated conversion tool, a good deal of network verification and editing with aerial photos will be necessary, especially for signal controllers, prohibited turns and other intersection details.

Conditioning, validation and calibration require detailed data. DTA differs from STA in effort required to produce meaningful results. Getting a network to clear can take considerable effort under high demand conditions, and may require demand matrix calibration. In our experience, there is no “low maintenance” DTA, because the results of a low-level effort are unlikely to be useful. For DTA to be a worthwhile endeavor, the user must be prepared to invest the necessary resources to achieve useful results, which will likely involve gathering plenty of high quality lane-detector data and turning-movement counts.

Find a suitable signal coding solution. Which level of signal collection effort is warranted and worth it? There are currently no guidelines on how to convert signal timing plans to the format needed for DTA and no evidence of how current DTA signal simplifications (no semi-actuated, no dual-ring) affect model results. For meaningful results in smaller study areas, developers agree that real signal timing plans should be used. For a regional DTA some form of synthetic signal timing might be necessary. Currently, there is no recommended practice for handling future-year signal timing.

Future year network must be created. Because DTA adheres strictly to network capacity, saturated demand (as in a future year scenario) quickly leads to system-wide gridlock, and thereby useless model results. To test future year demand, the DTA network must be coded with expanded capacity.

GUI preferences are important. The user will experience working with a DTA program primarily through its GUI. It is important, therefore, to use a DTA with an easy-to-work-with GUI, or a DTA that may be integrated with an external GUI of your choice.

Distribution and support are keys to success. Working closely with the DTA provider is important in the initial phases of network development and calibration. Good support and training is essential. As with any software, the user has to consider preferred program distribution (commercial or open source) and identify what level and intensity of support is likely to be needed, depending on internal staff resources.

Appendix A: Signal Timing Conversion

It is recommended that real-world signal timing plans be used for small study areas. Real signal timing operations can be rather complex, however, and DTA programs can currently only take simplified plans as input. This appendix describes in some detail different types of signal timing plans and the assumptions and simplifications that were made in this project.

Signal timing plans can be pre-timed, with fixed phases and cycle lengths, and phase rotation. Pre-timed signals are common in downtown areas. In actuated signals, the cycle length and order of phases constantly shift depending on demand at the different approaches. A mix of pre-timed and actuated approaches is very common. Multiple intersections in high-volume corridors can be coordinated (e.g. synchronized) at all times or during peak hours to create a “green wave.”

The study area has 135 signalized intersections, all of which are semi- or fully actuated. These locations are shown below in Figure A.1. Approximately 40 coordinated signals operate along prominent thoroughfares, such as Barnes, Cornell, Murray, Farmington and Canyon Roads. Three jurisdictions, City of Beaverton, Washington County and ODOT, operate the traffic controllers in the study area. The two types of controllers used are Northwest/Voyage by Northwest Signal Supply, and W4IKS by Wipiti IKS.

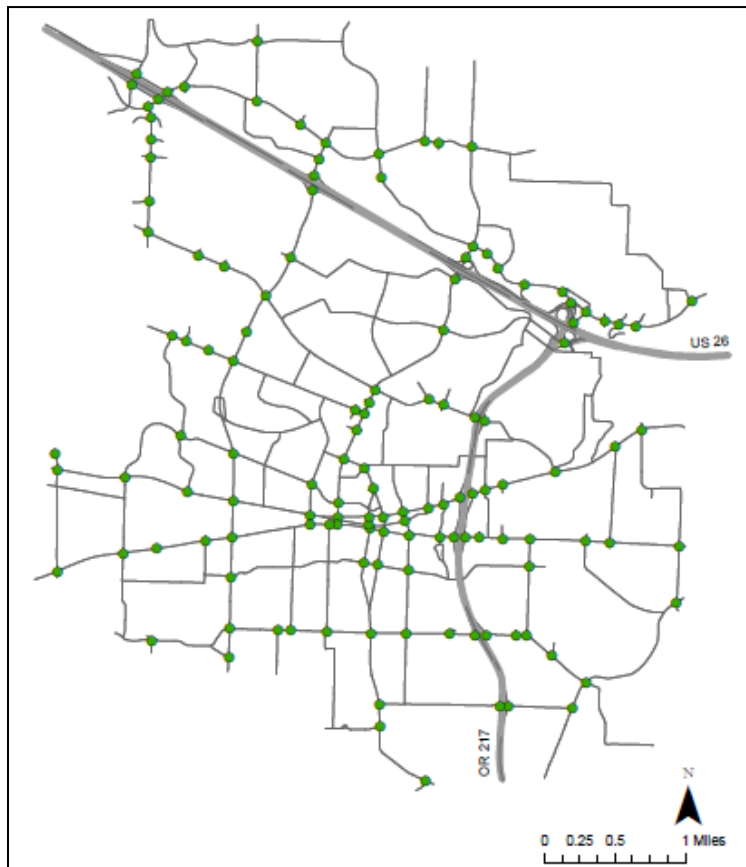


Figure A.1. Locations of signalized intersections in the study area

Signal timing plan information formats and code conventions differ between manufacturers. Actuated, uncoordinated signals often have two maximum green times, sometimes with a plan governing which green time is in effect at different times of day. Additional information on signal controllers may be available from the manufacturers. For example:

<http://www.oregon.gov/ODOT/HWY/SIGNALS/docs/2070info/VoyageVer21.pdf?ga=t>

Many intersections have different timing plans for peak and off-peak periods, and the exact times that timing plans switch varies by location. For simplicity we defined four signal timing periods:

- AM peak from 6:30 to 9:00
- Midday (Off-Peak 1) from 9:00 to 3:30 p.m.
- PM from 3:30 to 6:30 p.m.
- Evening and night (Off-peak 2) from 6:30 p.m. to 6:30 a.m.

The signals in the study area were then classified into one of four coding categories:

- **Case 1:** An intersection with ONE green time for the entire day, NO coordination.
 - Enter the same plan into AM, OP1, PM and OP2 text files.
- **Case 2:** An intersection with ONE green time for the entire day, and MANY coordinated periods.
 - Create different plans for the different times of day, with the same green time throughout, but offset only during coordinated periods.
- **Case 3:** An intersection with MANY green times, NO coordination.
 - Create different plans for different times of day, varying the green time.
- **Case 4:** An intersection with MANY green times, and MANY coordinated periods.
 - Create different plans for different times of day, varying green times and offsets.

DynusT can accommodate pre-timed coordinated signals as well as a simplified form of an actuated, uncoordinated signal. Dynameq requires all signals to be converted to pre-timed and can accommodate coordinated and uncoordinated signals. Cycle length is calculated automatically in Dynameq, but is an input in DynusT. Right-turns on red are allowed by default in DynusT, and can be allowed or prohibited in Dynameq.

Figure A.2, below, shows the GUI interfaces for entering pre-timed signal timing and phasing information for each program.

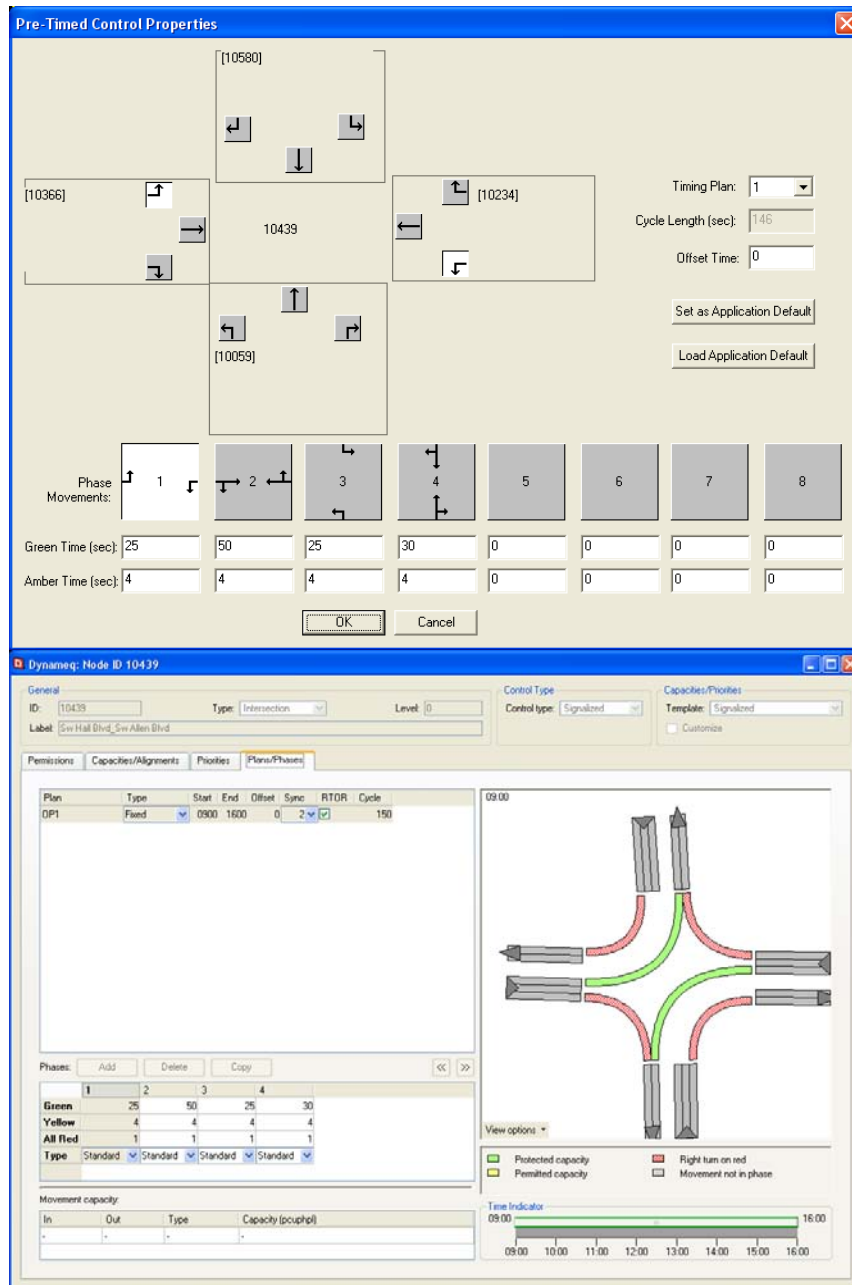


Figure A.2. Pre-timed signal controls represented in DynusT (top) and Dynameq (bottom)

DTA programs need information on phase rotation order. This is sometimes provided in the controller output sheets, but often needs to be decided using aerial photos or the judgment of the DTA modeler. When available, intersection diagrams like the one shown in Figure A.3, below, are very helpful for coding the order of phases. In a four-way intersection, the major approaches will generally get phases 2 and 6, i.e., the longest green times. A typical four-way intersection in the study area is 158th and Waterhouse. North and south are the major approaches. North and southbound left turns go in the first phase. The second phase is north and southbound through and right turn movements. The third phase holds the westbound left, through and right movements, and phase four accommodates all eastbound movements.

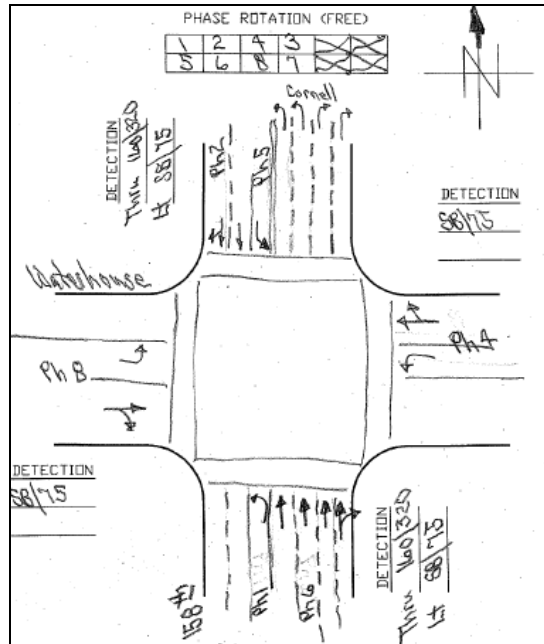


Figure A.3. Signal phasing diagram for a typical 4-way intersection.

Three-way intersections typically have three phases, an example which is shown below in Figure A.4. First goes the major movement that also includes a left turn, in this example eastbound left and through movements. Next goes the opposing major movement, here westbound through and right. Eastbound through continues through the second phase, as there is no conflicting movement. The third phase holds minor approach movements, here southbound left and right.

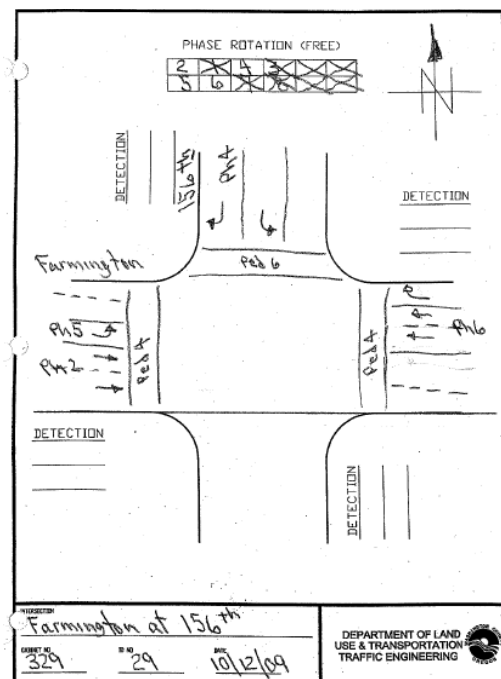


Figure A.4. Example of a signal phasing plan for a 3-way intersection

Often, two movements that go together have unequal green times, but Dynameq and DynusT require that all movements going during the same phase have the same amount of green time (and amber and red). If the difference is large, it may be worth creating an extra phase to accommodate the additional green time of a movement. In our study, if the difference was a matter of five or ten seconds, the longest phase was typically chosen, as minimizing the number of phases is typically preferable.

Cardinal directions can be ambiguous, as shown in Figure A.5, below. In the top picture, Hall Boulevard is the major approach. In the VISUM network, Hall Boulevard was coded as a north and south approach. The signal timing sheet has Hall Boulevard as an east-west approach. To avoid having to recode movements in the DTA, the signal timing sheet information was rotated 90 degrees, so that the east-west timing was given to the north-south approach. In other cases, such as depicted in Figure A.6, below, directional ambiguities combined with more than four turning movements necessitated coding multiple approaches of the same cardinal direction.

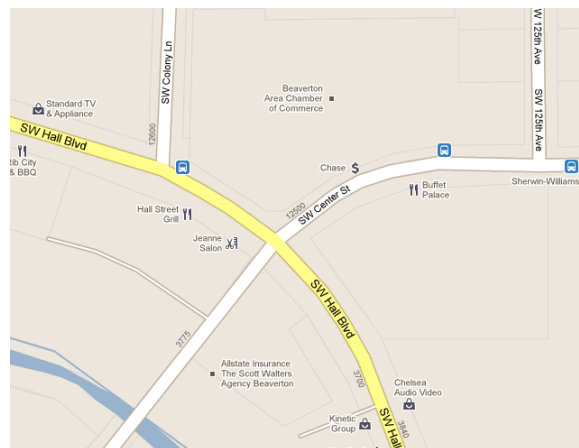


Figure A.5. At this intersection, cardinal directions of approaches are ambiguous.

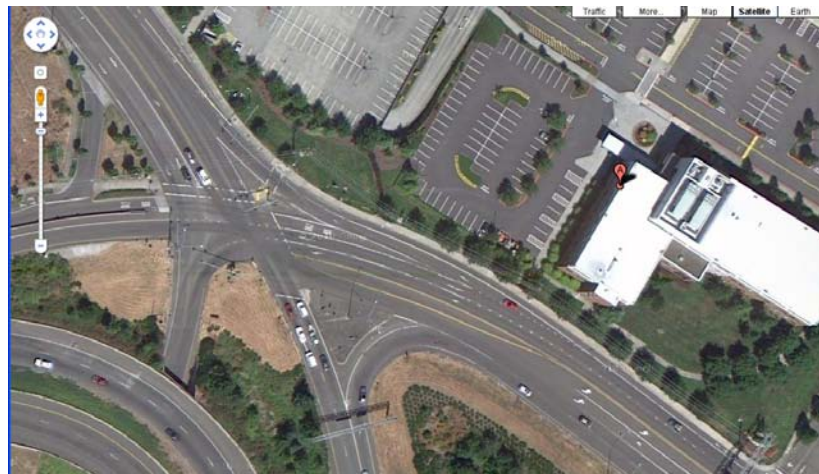


Figure A.6. Barnes Road at the intersection of US 26 and OR 217

Volumes and pedestrian movements can determine the coding of a left turn as protected or permitted. A left turn can shift between protected and permissive during the same phase. Permitted left turns are given lower capacities in DTA, but they must be simplified. In this study, left turns were coded as permissive only when sharing a phase with an opposing through/turn movement of vehicles, but pedestrian movements were not considered. When not apparent from the signal timing documents, aerial photos were consulted to decide whether movements were protected or permitted. Synchro files were converted to Dynameq input by INRO. With a couple of exceptions at external stations, as mentioned above, all movements coded as permitted in Synchro files were retained as permitted, even when the coding was caused by pedestrian crossings rather than opposing vehicular traffic.

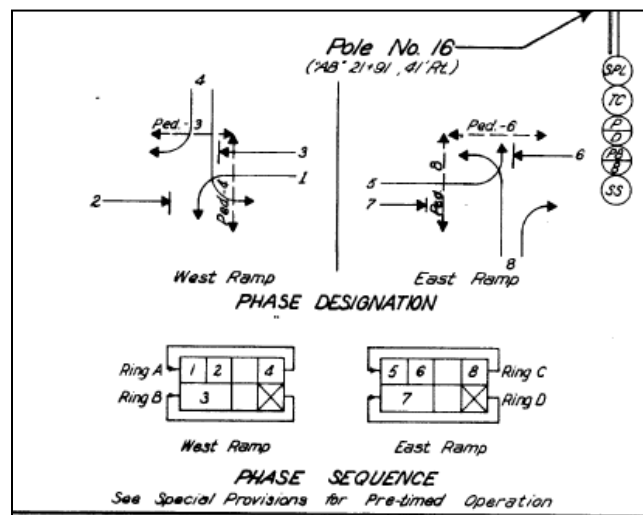


Figure A.7. Phasing diagram for a controller governing signal timing for two intersections at a freeway interchange

At a few intersections at freeway interchanges, the on- and off-ramps connecting to an overpass road on both sides of the freeway are controlled by a single controller. Figure A.7 shows an example phasing diagram for eastbound and southbound Allen Boulevard and OR 217.

For coordinated signals, the DTA programs take as input green, amber and red time as well as synchronization phase and offset reference point (e.g. synchronization phase). Offset reference point differs between signal timing plans and the DTA programs. In Synchro, for example, the reference point is the beginning of the amber phase; in Dynameq it is the beginning of green phase; and DynusT uses the end of the green phase.

Figure A.8, below, illustrates the calculation of the offset in Dynameq from a Synchro plan. Phase two is the major approach. Starting with a given offset (in seconds), the yellow and red time of the second (reference) phase is added, plus the full green, yellow, red times of all phases following phase two. In DynusT, an offset is specified for the first phase of each cycle relative to the start-time of the analysis (zero seconds). The reference phase needs to be set as the first phase of the timing plan.

0	72	77	108	122
Given offset	Y+R of 2 nd phase	31 sec split of 3 rd phase	14 sec split of 1 st phase	Start of green of 2 nd phase

Figure A.8. Example calculations used to convert timing plan offsets for use in Dynameq

References

- Chiu, Y-C, J. Bottom, M. Mahut, A. Paz, R. Balakrishna, T. Waller and J. Hicks. A primer for dynamic traffic assignment. ADB30 Transportation Network Modeling Committee, Transportation Research Board, 2010. Accessed from http://www.fsutmsonline.net/images/uploads/mtf-files/dta_primer.pdf.
- Dowling, R., A. Skabardonis, and V. Alexiadis. Traffic Analysis Toolbox III: Guidelines for Applying Traffic Microsimulation Software. Publication No. FHWA-HRT-04-040, USDOT, Federal Highway Administration, July 2004.
- Hao, J., M. Hatzopoulou and E. Miller. Integrating an activity-based travel demand model with dynamic traffic assignment and emission models, Transportation Research Record: Journal of the Transportation Research Board, No. 2176. TRB, National Research Council, Washington, D.C., 2010.